

Wizualizacja danych

Grafy, drzewa i sieci



Jerzy Stefanowski
Instytut Informatyki
Politechnika Poznańska

Wykład TWO - aktualizacja 2017/18

Plan wykładu

Przykłady prezentacji grafów, drzew i sieci
(także eksploracja giga-danych i WWW)

Sieci społeczne

Definicje z teorii grafów

Rysowanie grafów – podstawy „Layout algorithms”

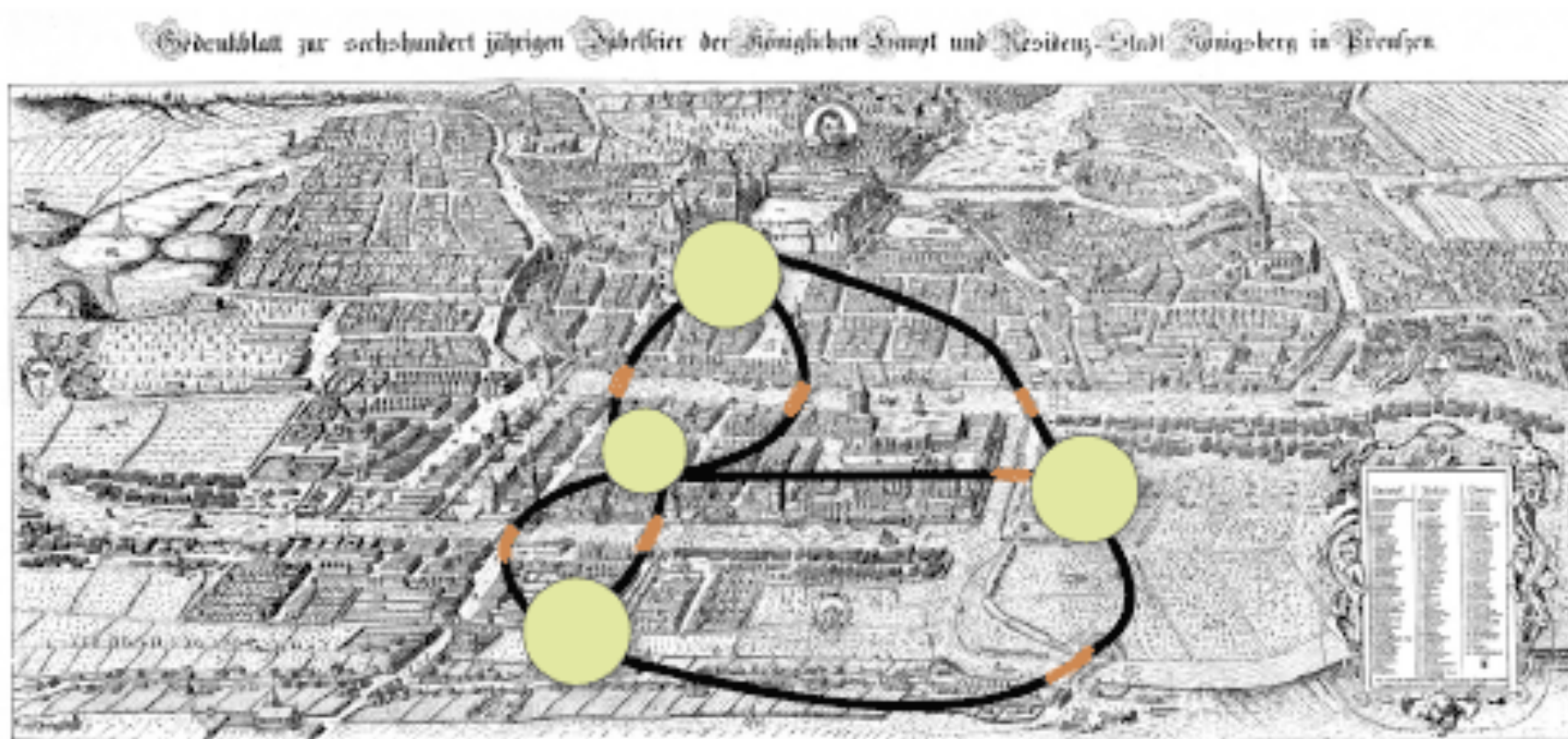
1. Kryteria i miary oceny
2. Popularne układy struktur grafowych
3. Podstawowe algorytmy (Algorytmy sił fizycznych oraz kołowe)

Interakcja i nawigacja w bardzo dużych grafach i sieciach

Drzewa – różnice

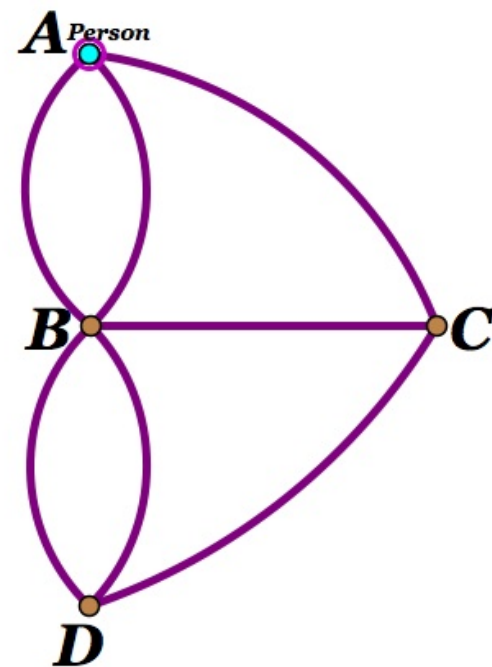
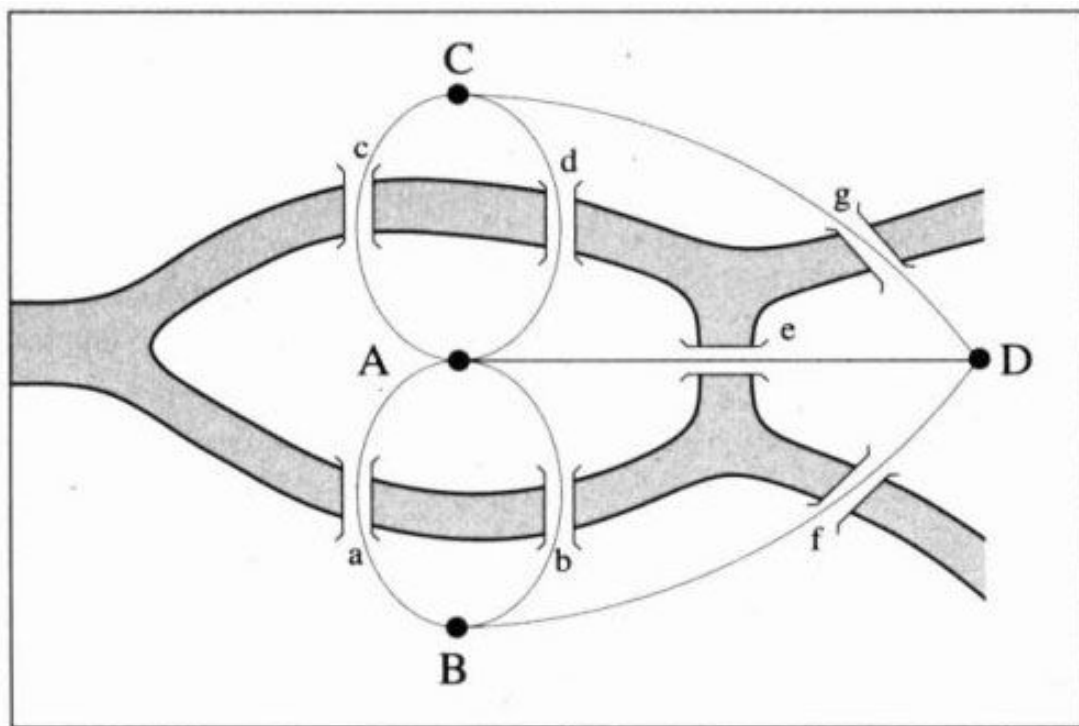
1. Podstawowe algorytmy
2. Przykłady wizualizacji

Grafy – historyczne uwarunkowania



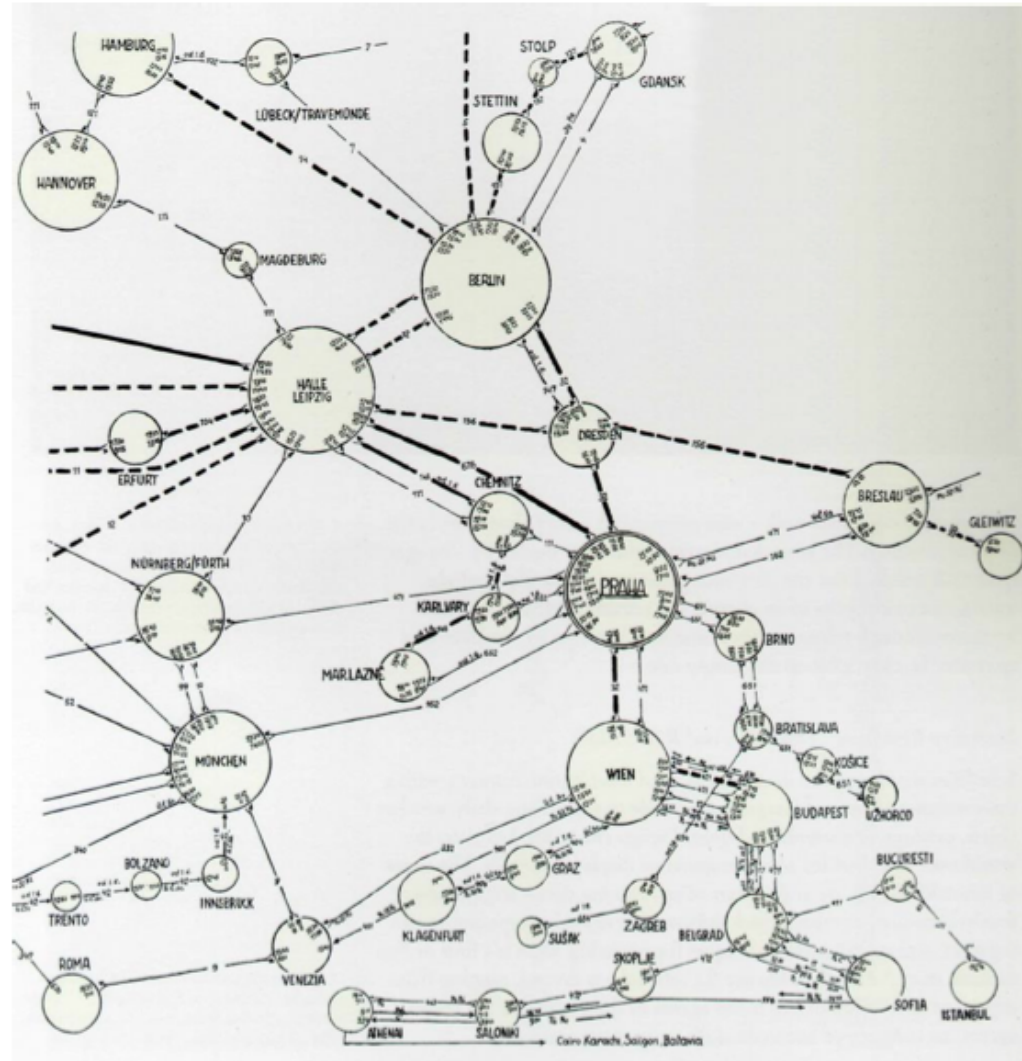
Spacer po mostach w Królewcu

Grafy – problem mostów w Królewcu



Euler – model grafu

Struktury sieciowe / grafy w wizualizacji informacji (lata 30te)



Rysunek 1 Plan lotów na terenie Czechosłowacji z roku 1933. Źródło: Edward Tufte, Envisioning Information, Connecticut: Graphics Press, 1990, s.102

Analiza sieci - przykład rozwoju Internetu

ARPANET 1971



MAP 4 - September 1971

ARPANET 1980



Sources:

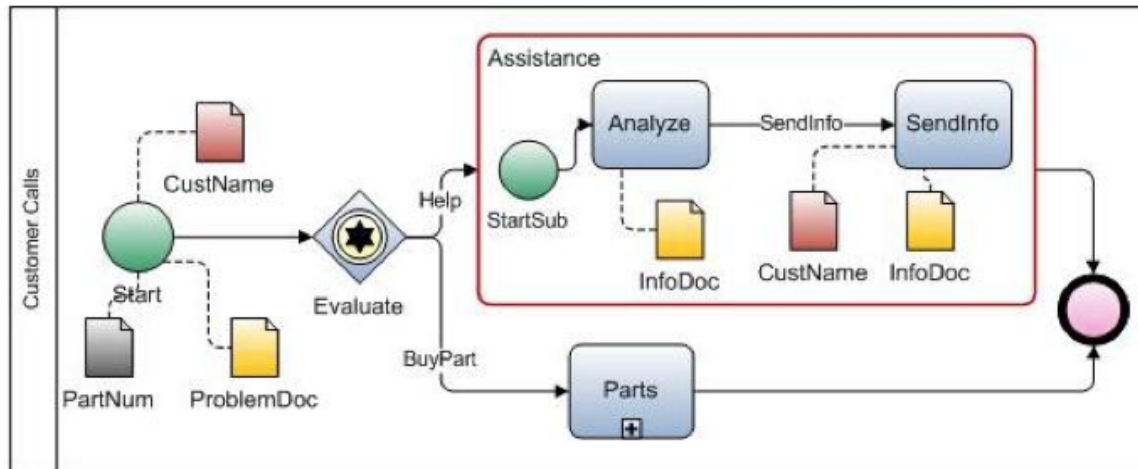
1971 - "Casting the Net", page 64;

1980 - http://mappa.mundi.net/maps/maps_001/

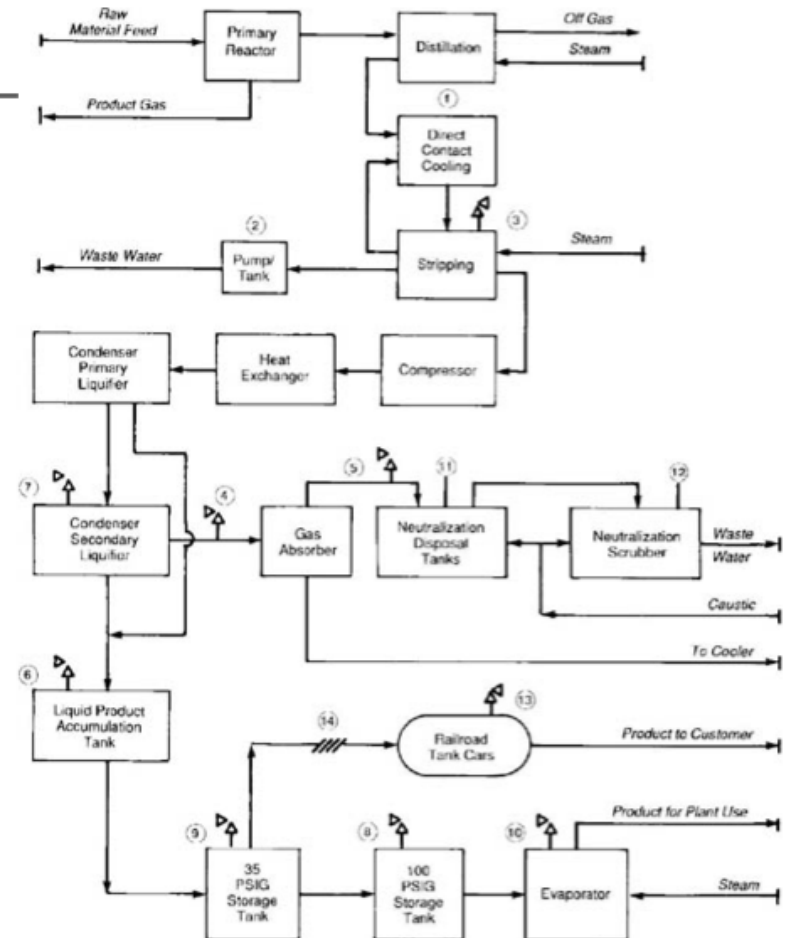
<http://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html>

Modelowanie i analiza procesów biznesowych

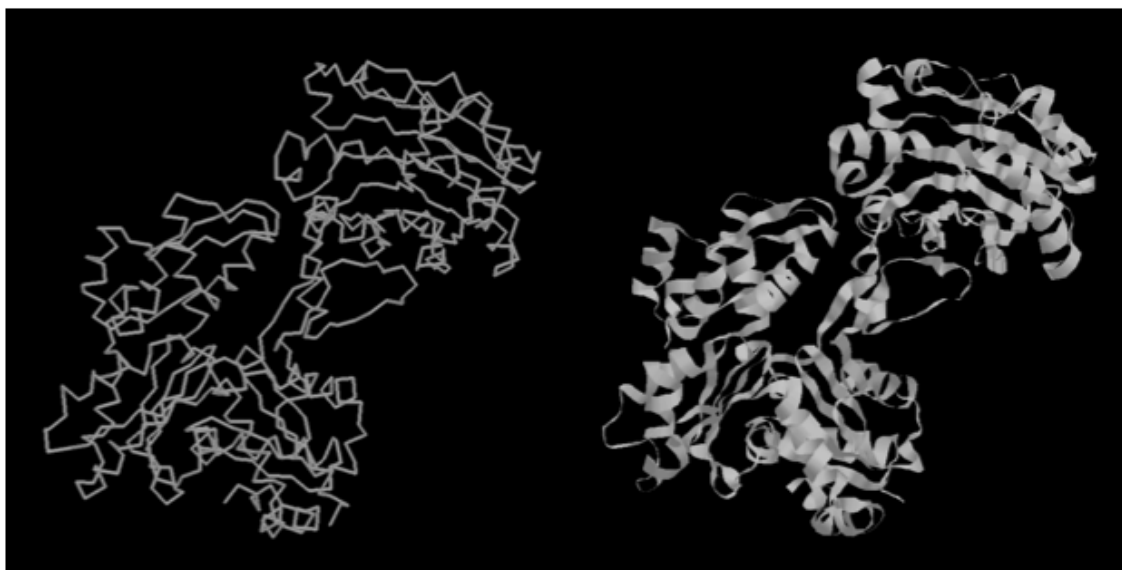
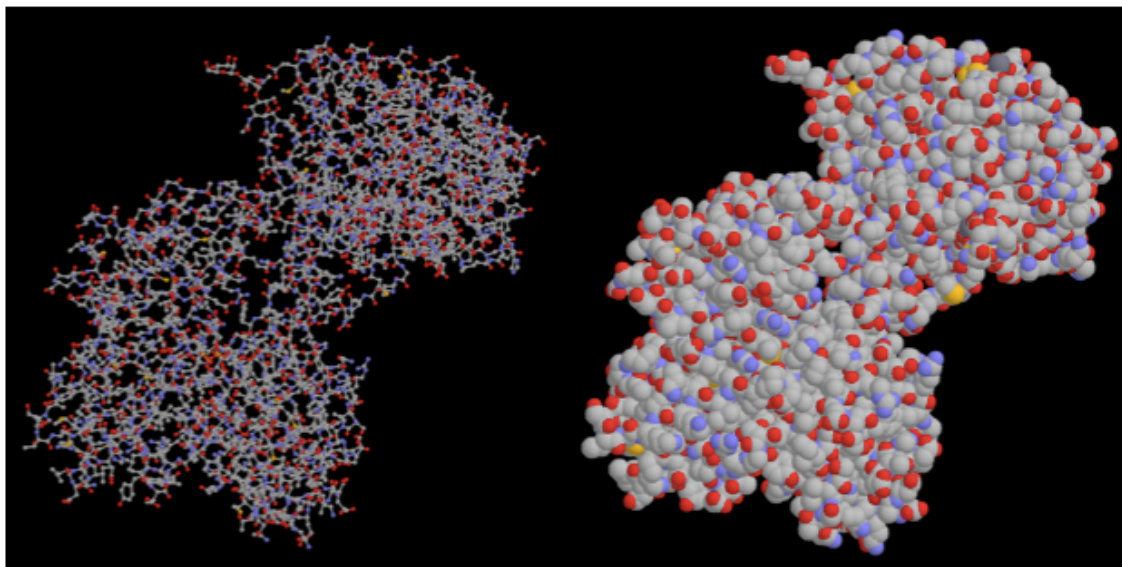
Process flow diagrams



Page-1

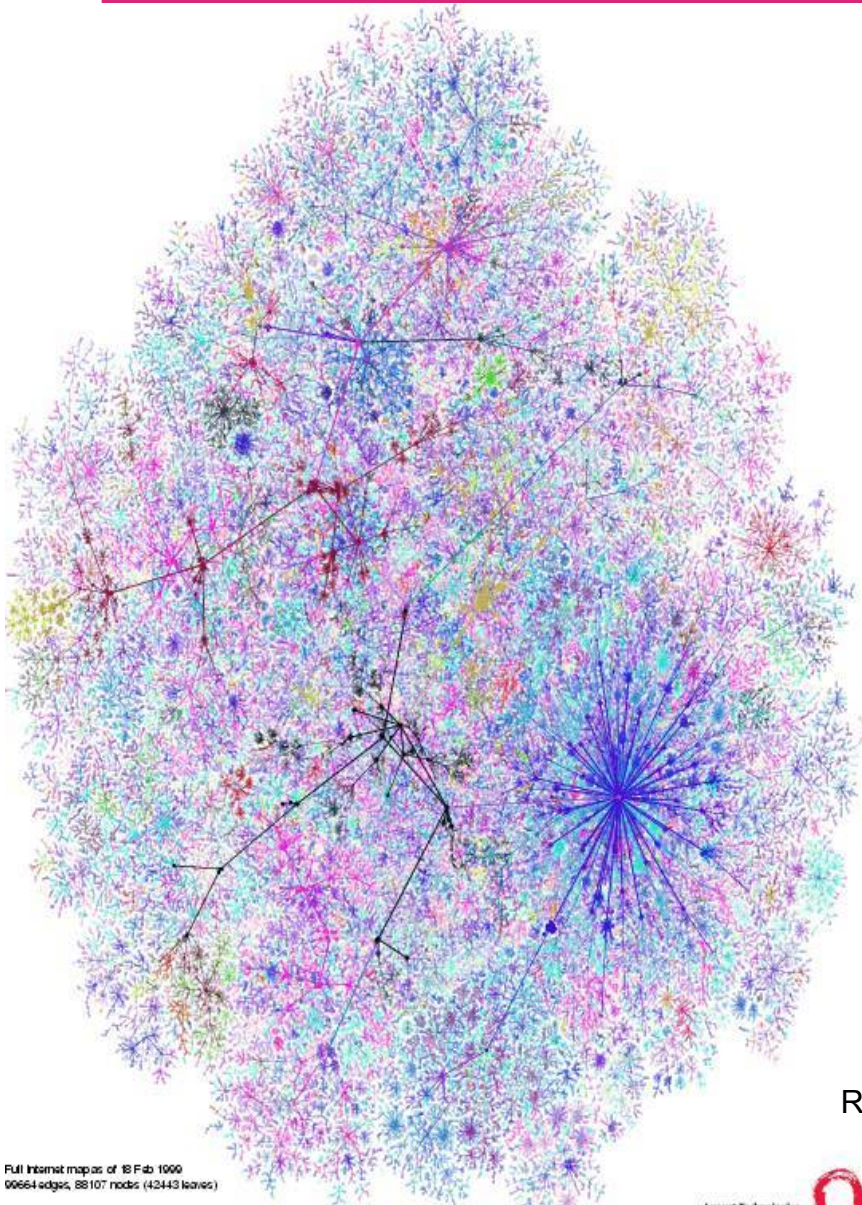


Wizualizacja struktur biologicznych i chemicznych

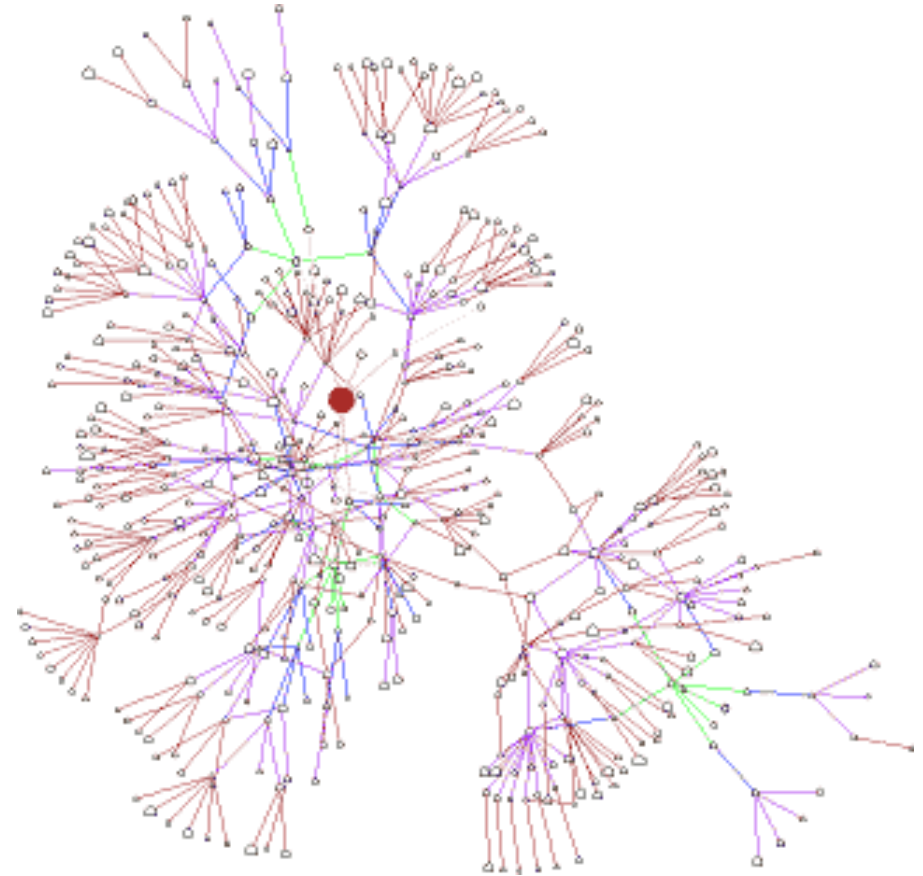


Za V.Batagejl ULjubljana

Obecnie rozważane grafy i sieci (Internet)



Routerzy i iserwery

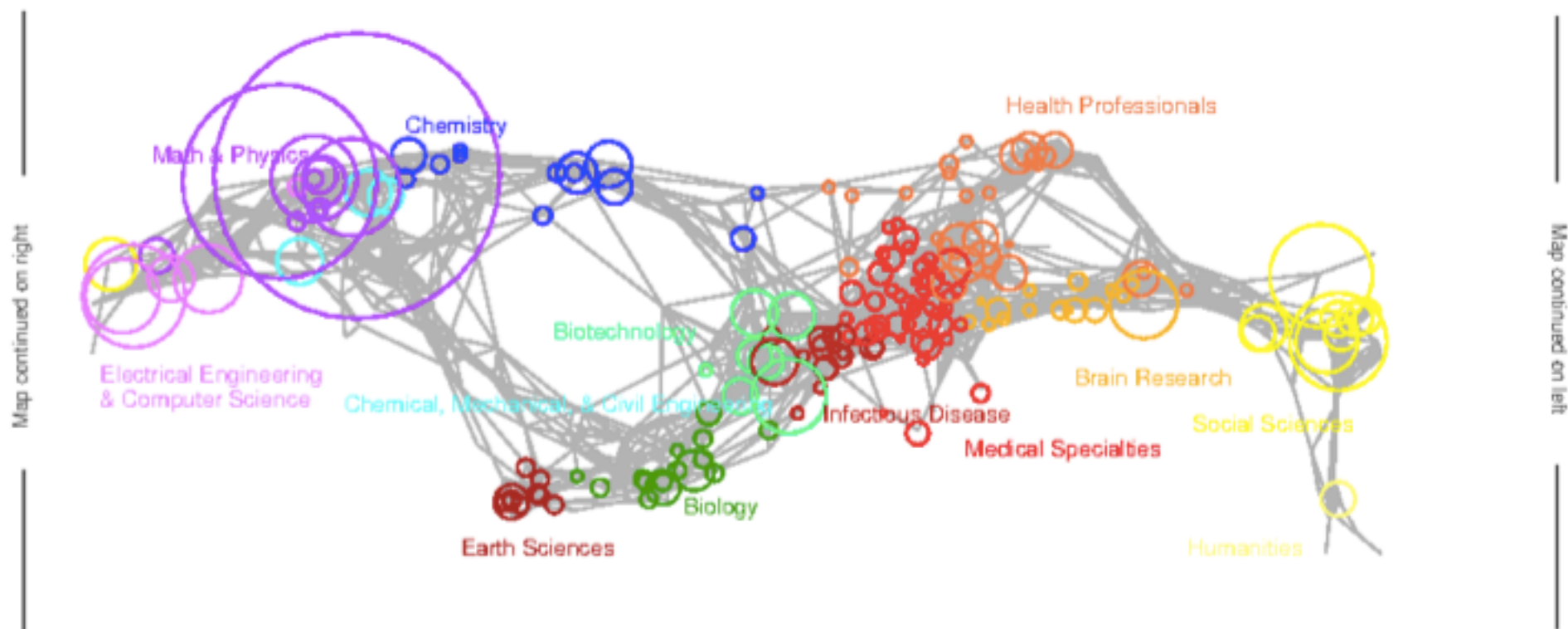


Sieć społeczna
Ważne relacje między węzłami

Grafy i sieci (analiza tekstów i tematyki)

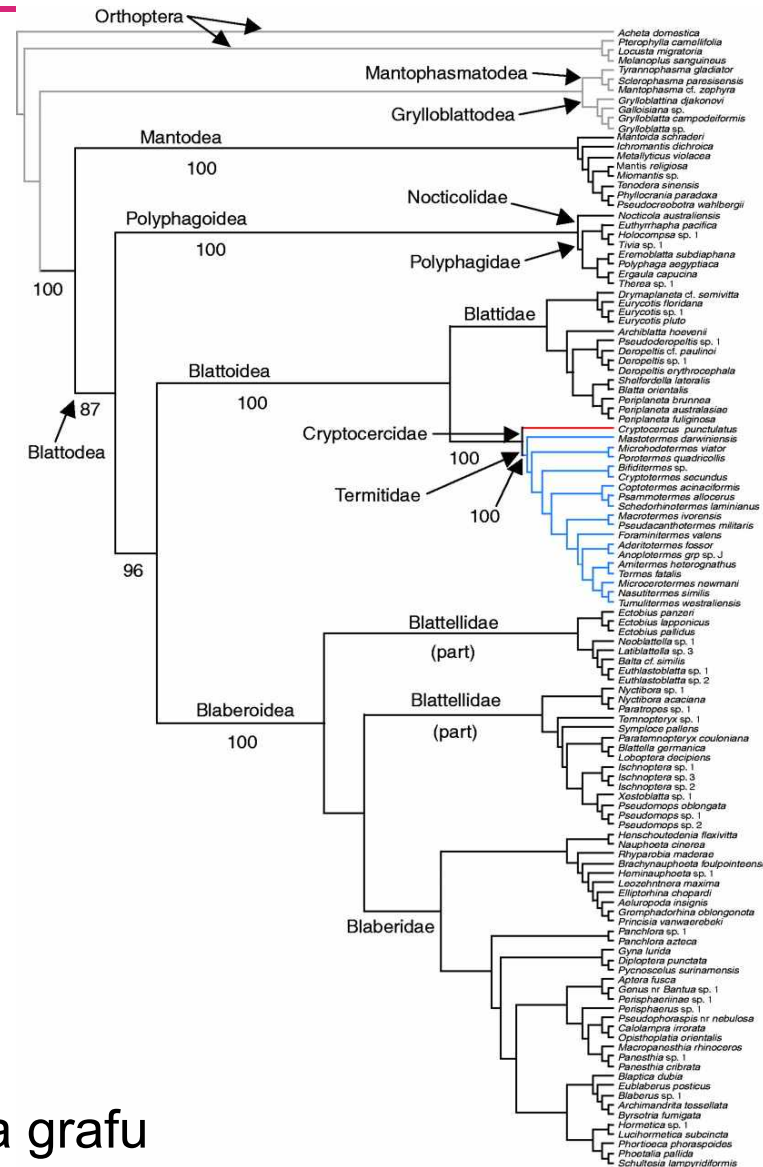
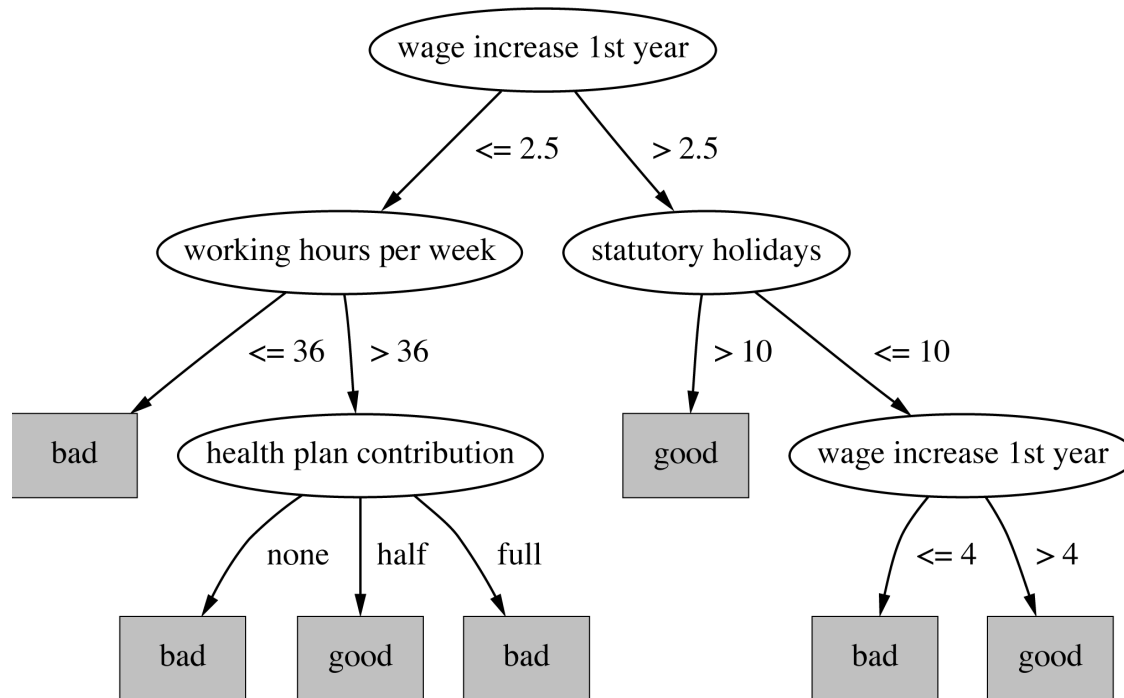
Topical Visualization

Generated from ISI Data: D:\sci\sciz\sampladata\scientometrics\vis\FourNetSciResearchers.isi
90 out of 112 records were mapped to 182 subdisciplines and 13 disciplines.
maj 04, 2014 | 08:28 PM CEST



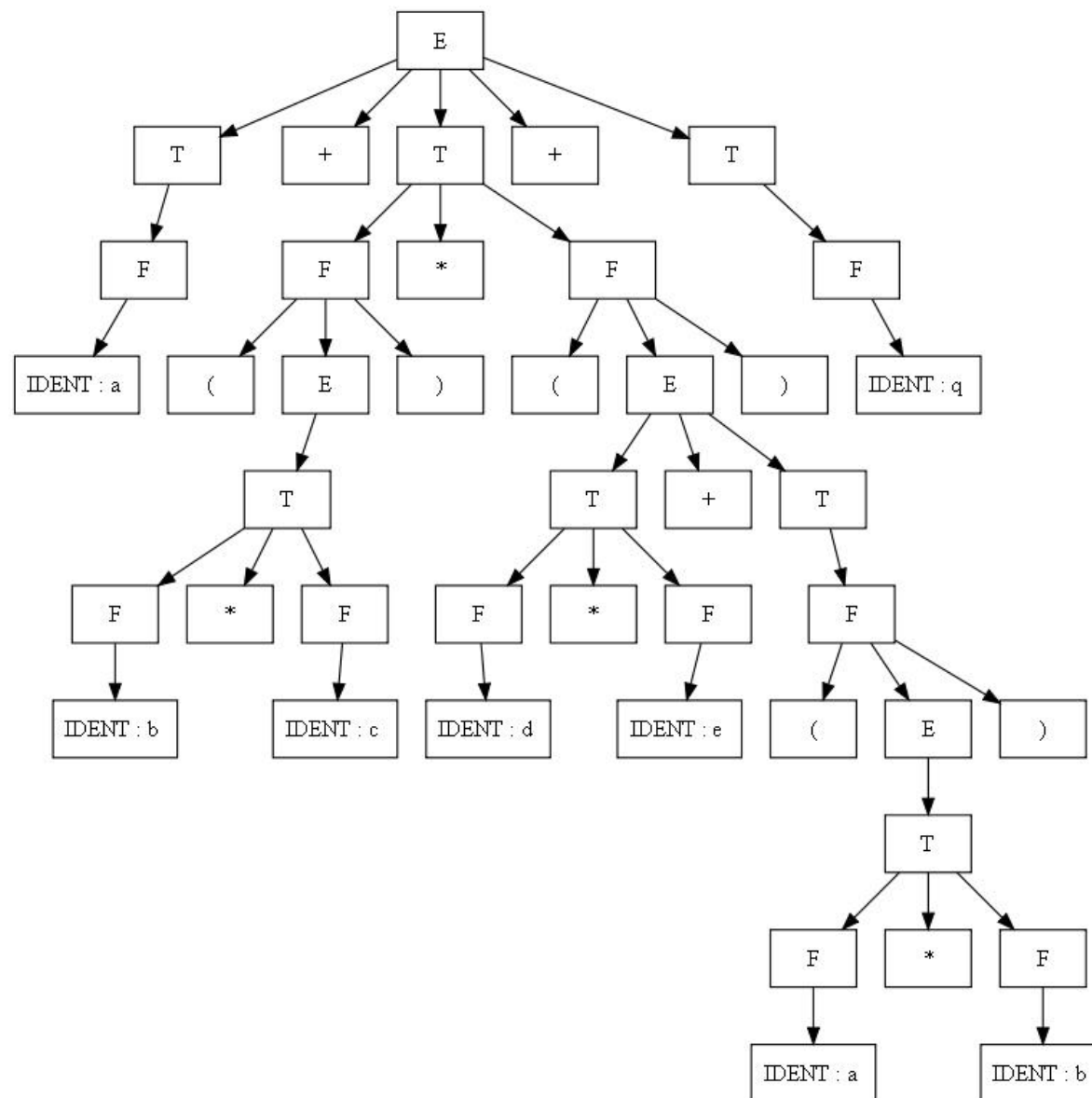
2008 The Regents of the University of California and SciTech Strategies.
Map updated by SciTech Strategies, OST, and CNS in 2011.

Wizualizacje w postaci struktur drzew



Struktura hierarchiczna – popularna odmiana grafu

Drzewo wizualizujące gramatykę języka programowania

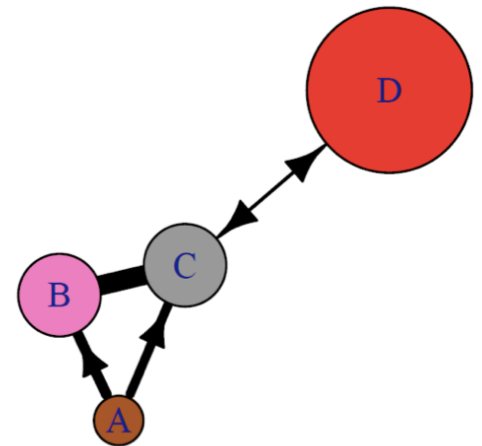


Grafy wykorzystywane w wielu dziedzinach

- Obrazowanie struktury zasobów
 - Katalogi, pliki na dysku (drzewo)
- Taksonomie w nauce (drzewa klasyfikacyjne)
 - Biologia, chemia i inne nauki
 - Mapy bio-genetyczne
 - Organizacja bibliograficzna
- Schematy organizacyjne
- Modelowanie procesów biznesowych (workflow diagrams)
- Zarządzanie projektami (CPM, PERT)
- Zapis algorytmów lub modeli danych (UML, DB)
- Wyszukiwanie informacji
- Konstruowanie schematów logicznych (VLSI)
- Analiza stanów maszyn (automatyka, informatyka)
- Socjologia (sieci społeczne)
- Eksploracja i nawigacja w Internecie

Kilka pytań

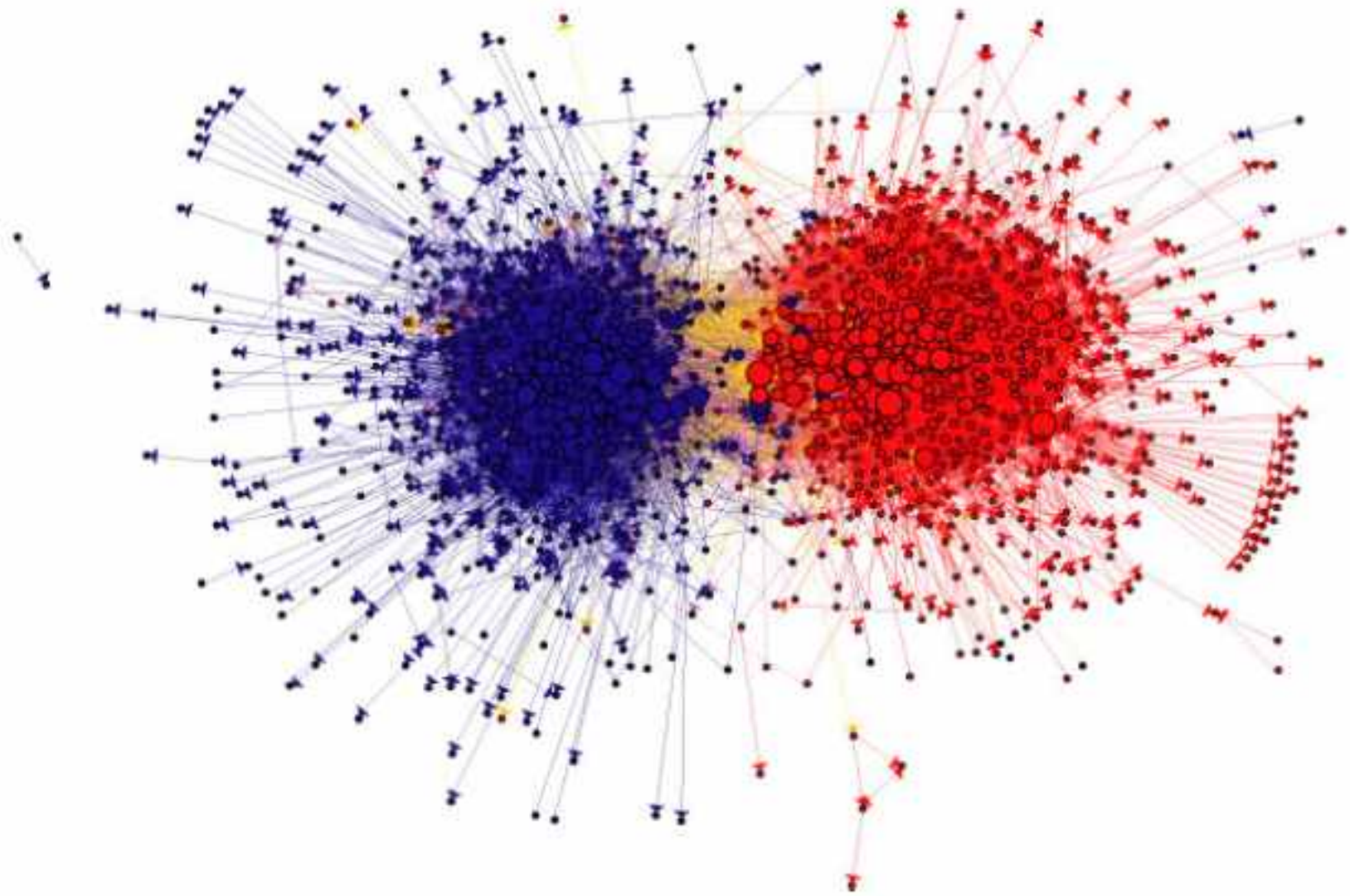
- Czy grafy i wizualizacje sieci są czytelne?
- Jak je rysować?
 - Typowo tzw. node-link diagrams
 - Problemy reprezentacji grafów
- Czy można to algorytmizować?
 - Czy w pełni automatyczne?
 - Zależność od rozmiarów i rodzaju grafów
- Czy to jest proste zadanie?



Node-link diagram:

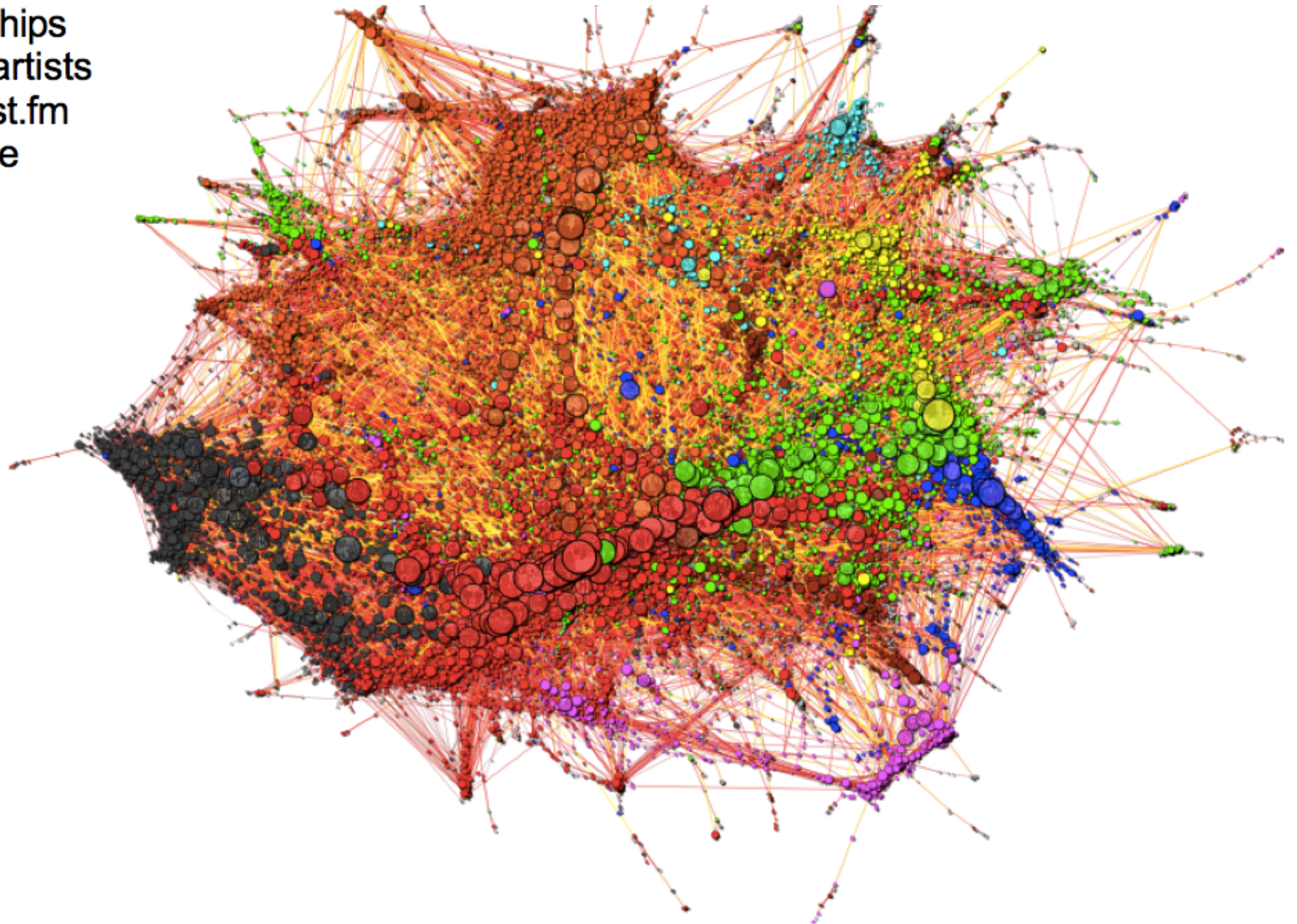
"When you think of a graph, you likely already think of a node-link diagram

- unless you're a mathematician." Kosara (2012)



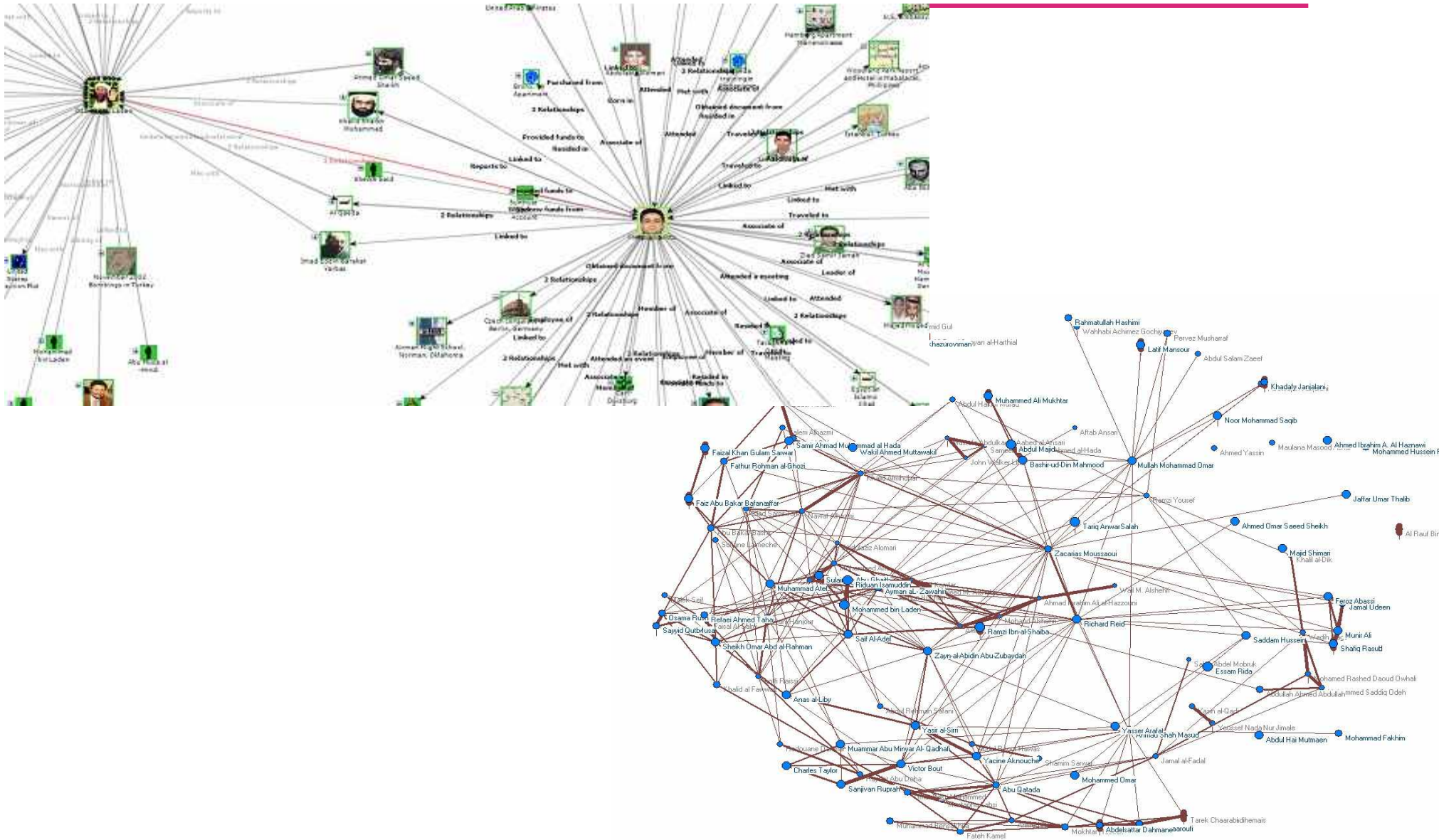
source: Adamic and Glance, [The political blogosphere and the 2004 US election: divided they blog](#), Proceedings of LinkKDD, Chicago, IL, p.36-43, 2005.

relationships
among artists
in the last.fm
database

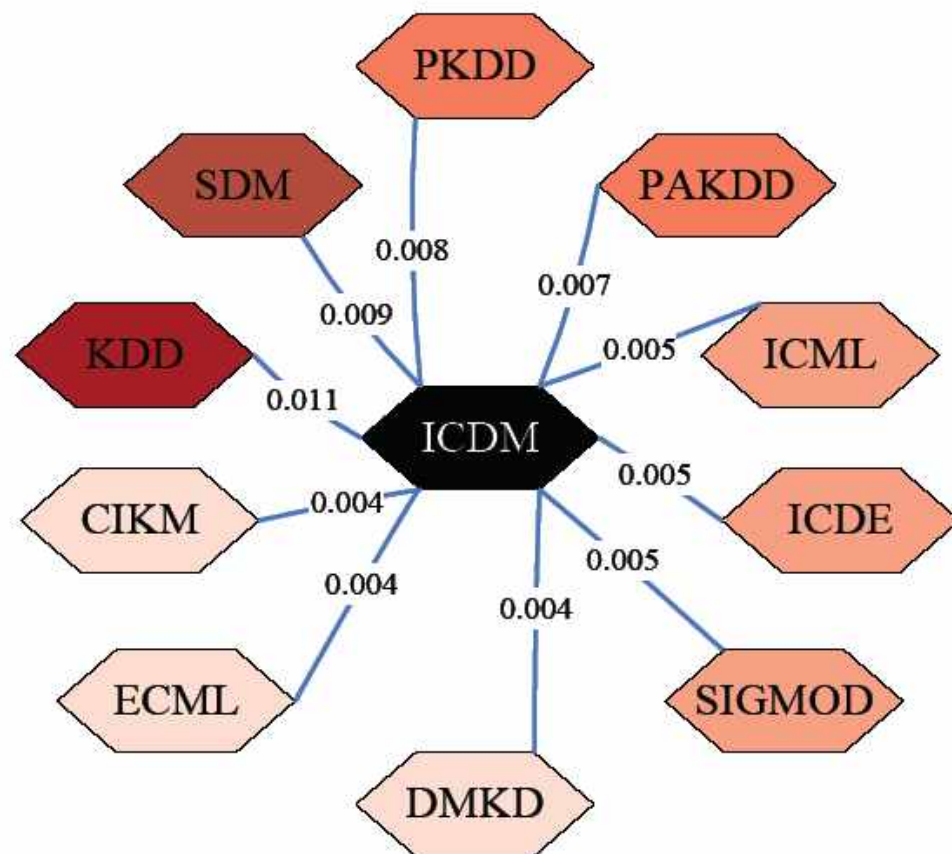
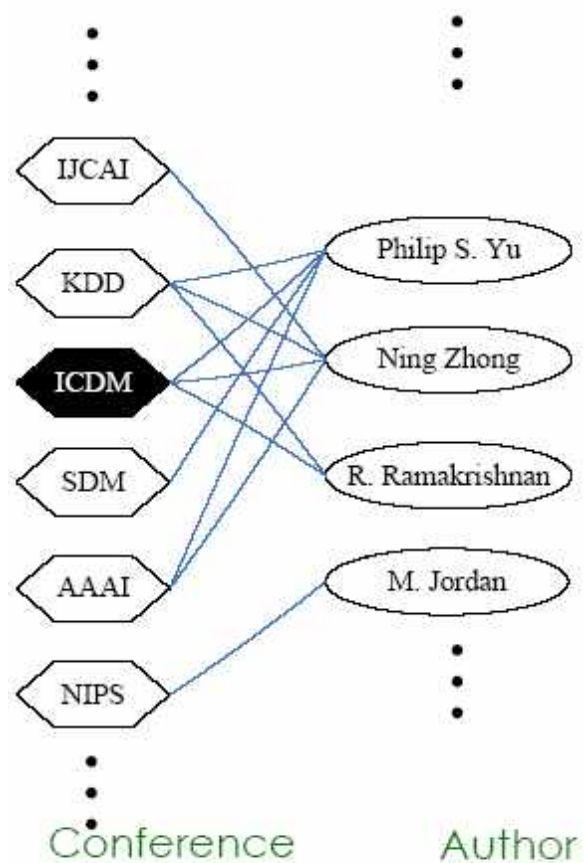


<http://drewconway.com/zia/2013/3/29/the-value-of-edges-in-complex-network-visualization>

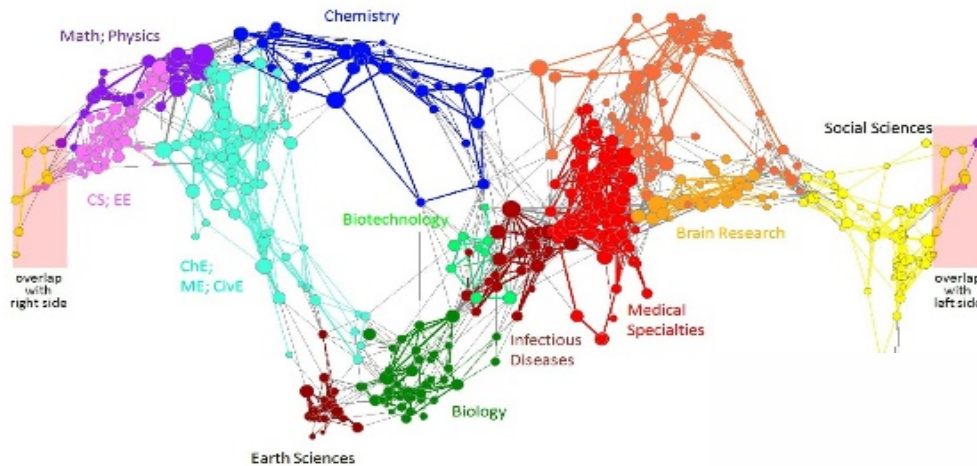
Analiza powiązań użytkowników Internetu



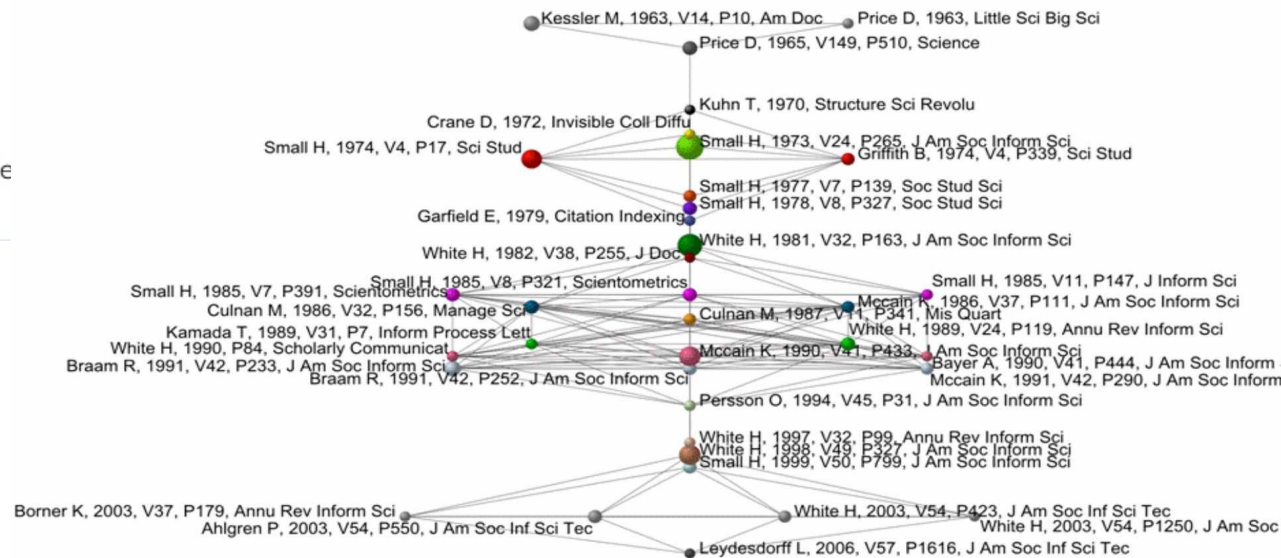
Analiza publikacji konferencyjnych



Mapy rozwoju nauki i inspiracji

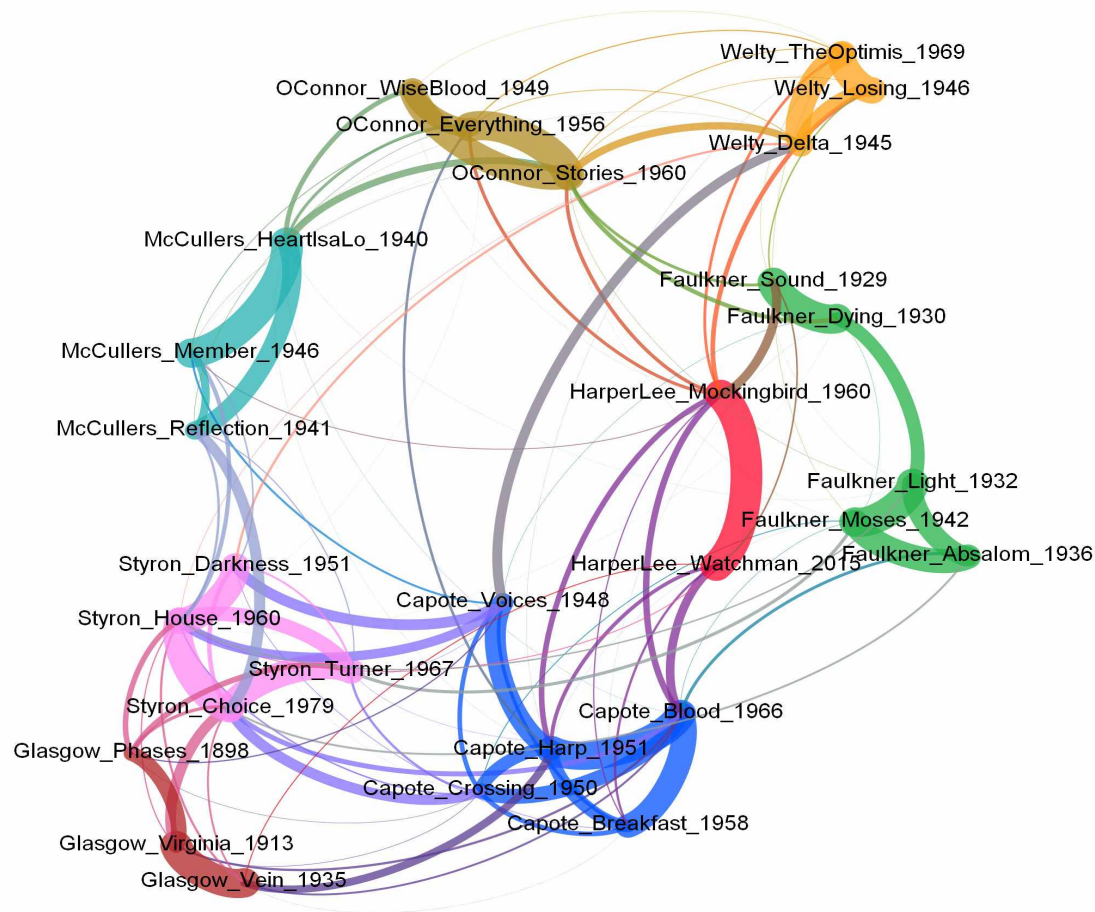


Citation networks and Maps of scie
[Börner et al., 2012]



Analiza podobieństwa tekstów amerykańskich pisarzy

- NLP, powiązania i inspiracje
- “Śledztwo literackie” - M.Eder i J.Rybicki 2015 - poszukiwanie podobieństwa stylu pisania słynnych książek
- Opis oceny autorstwa książki Harper Lee
 - Zabić drozda (1960)
 - “Go Set a Watchman” (2015)
 - Czy jest autorką, czy widać “odciski” innych osób w stylu tekstu?
- Więcej - artykuł Gazeta Wyborcza (31 07 2015)



Spójrz na blog BIQDATA.pl

Podstawowe definicje

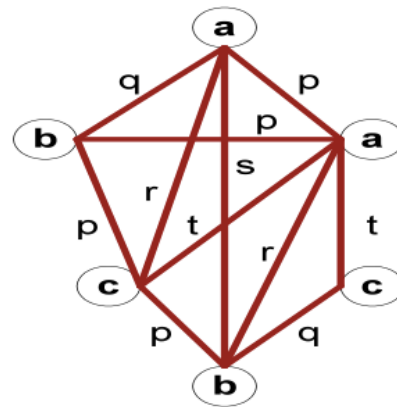
Graf $G=(V,E)$

- V – zbiór wierzchołków, E – zbiór krawędzi

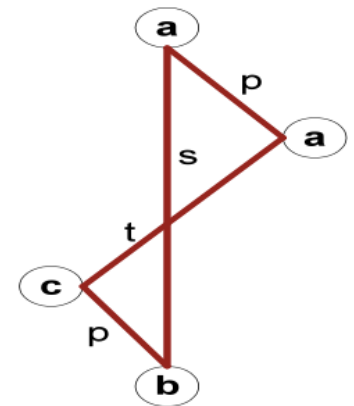
Grafy etykietowane

Krawędzie i / lub węzły opisane dodatkową informacją (etykietami) reprezentującymi ich właściwości

Graf $G'=(V',E')$ jest podgrafem $G=(V,E)$ jeżeli V' jest podzbiorem V oraz E' jest podzbiorem E



(a) Labeled Graph



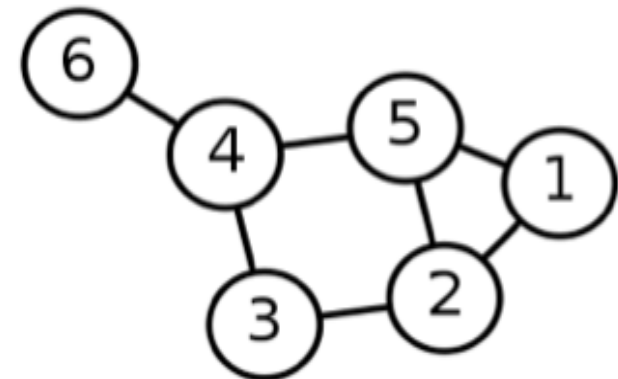
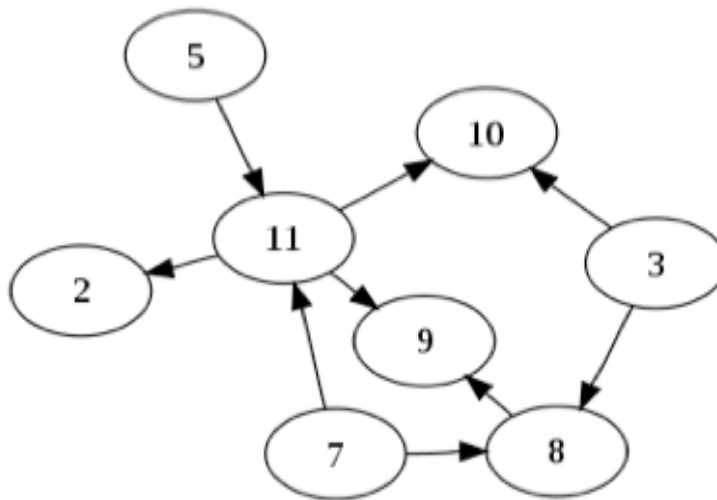
(b) Subgraph

Grafy skierowane i nieskierowane

Graf jest skierowany (ang. Directed) jeśli jego krawędzie są „zorientowane” (ozn. najczęściej strzałką)

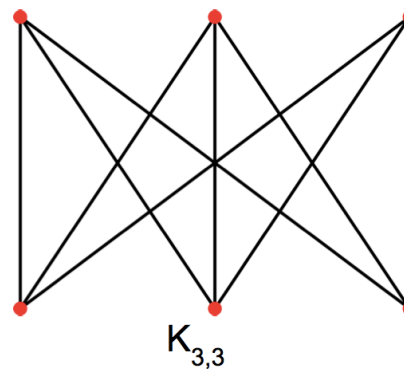
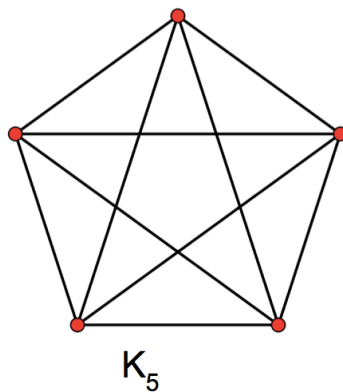
- Krawędź (u,v) jest innym obiektem niż (v,u)

Graf nieskierowany – krawędź (u,v) jest tym samym obiektem jak (v,u)

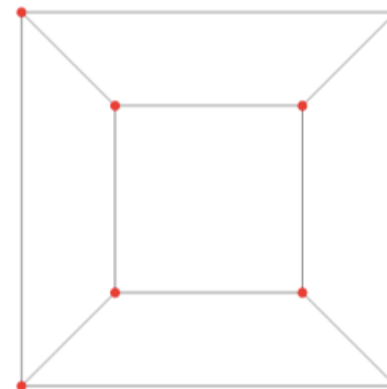


Planarność grafów

- Graf jest planarny, jeśli możliwe jest takie jego przedstawienie na płaszczyźnie, żeby krawędzie grafu nie przecinały się
- Graf jest planarny wtedy i tylko wtedy, kiedy nie zawiera podgrafów będącym grafem rozszerzonym K_5 lub $K_{3,3}$



- Grafy planarne są potencjalnie czytelniejsze



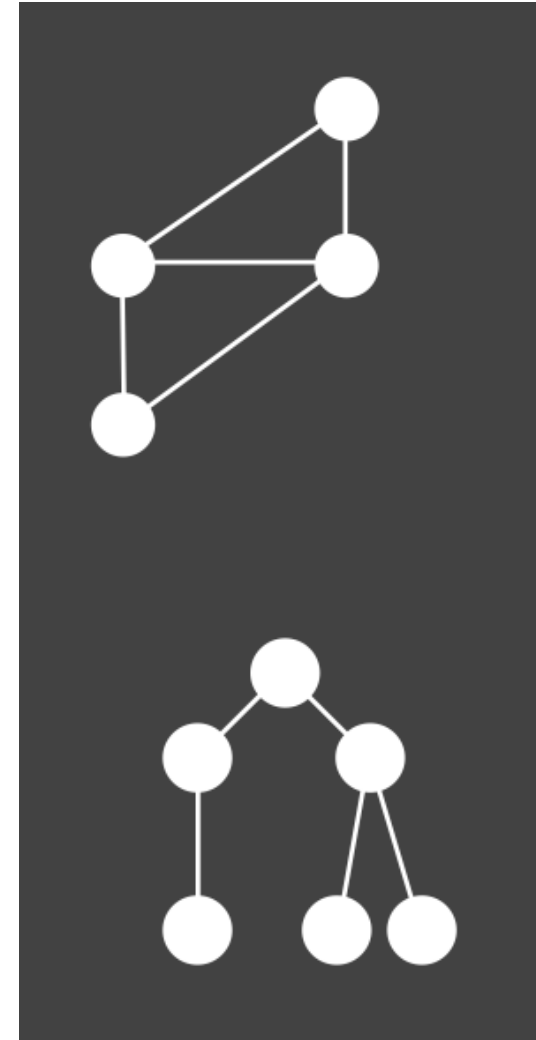
Grafy a drzewa

Graf

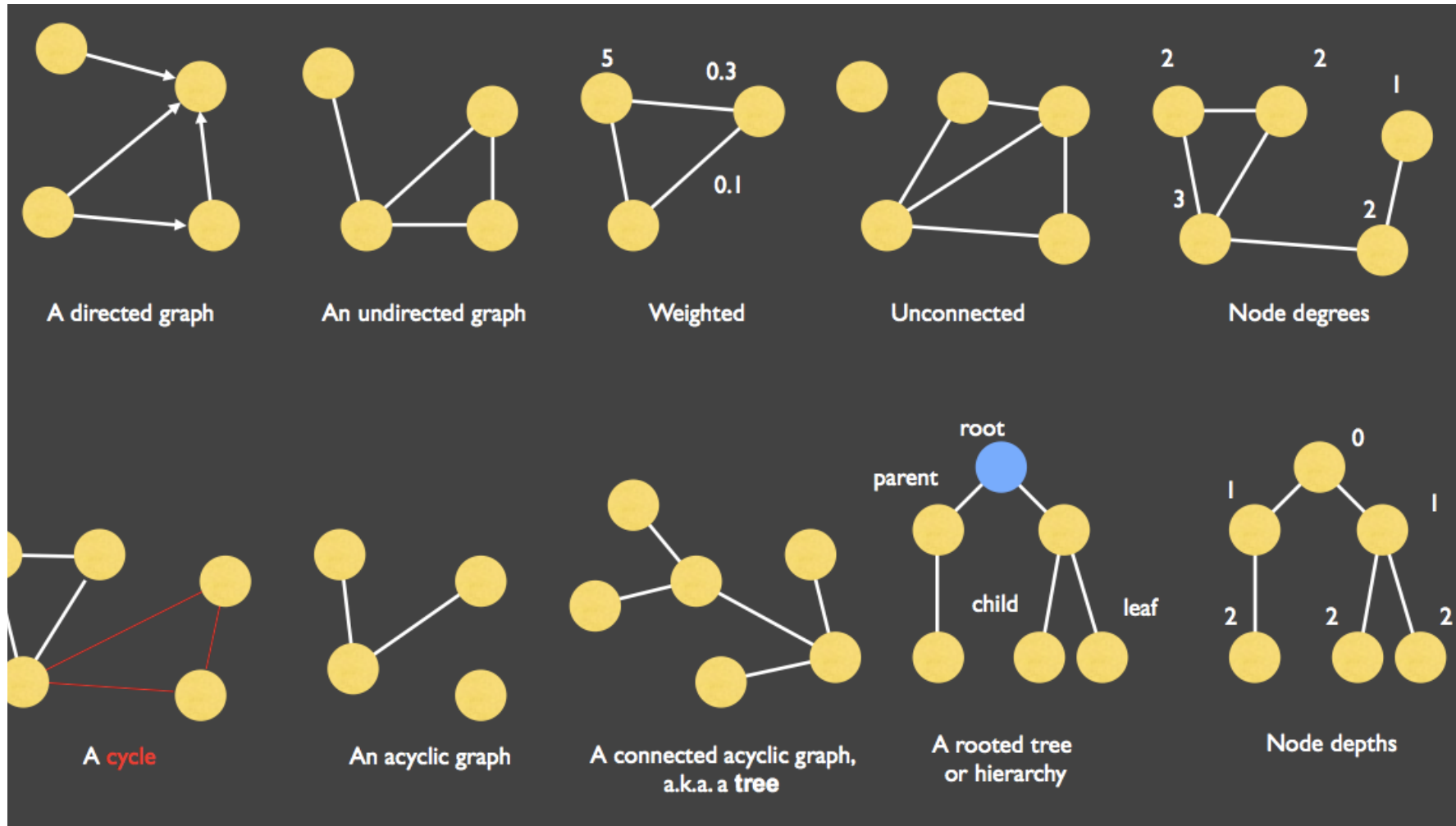
- Modeluje relacje między wieloma elementami danych
- Wykorzystuje różne powiązania węzłów i krawędzi

Drzewo

- Hierarchiczna struktura spójnego grafu acyklicznego z wyraźnym węzłem początkowym (korzeń drzewa)
- Korzeń, węzły potomne, liście



Różne rodzaje grafów



Reprezentacja grafów

Graf $G=(V,E)$

- V – zbiór wierzchołków,
- E – zbiór krawędzi

Typowo stosowane:

- Macierz sąsiedztwa (adjacency-matrix)
- Listy sąsiedztwa (adjacency-list)
- Lista krawędzi
- Macierz incydencji

Dogodność reprezentacja dla rodzaju grafu

Ocena – koszty vs. przydatność do wizualizacji

```
graph = [  
  [0, 1, 0, 0],  
  [0, 0, 1, 0],  
  [1, 0, 0, 1],  
  [0, 1, 0, 0]] # wagi (lub bity 0/1)
```

```
graph = [[1], [2], [0, 3], [1]] # bez wag (styl C++)
```

```
graph = {0: [1], 1: [2], 2: [0, 3], 3: [1]} # bez wag (styl GvR)
```

```
graph = {0: set([1]), 1: set([2]), 2: set([0, 3]), 3: set([1])} # bez  
wag (zbiory)
```

```
graph = {0: {1: 1}, 1: {2: 1}, 2: {0: 1, 3: 1}, 3: {1: 1}} # wagi
```

```
Graph=[(0, 1), (1, 2), (2, 0), (2, 3), (3, 1)] # bez  
wag
```

```
Graph=[(0, 1, 1), (1, 2, 1), (2, 0, 1), (2, 3, 1), (3,  
1, 1)] # wagi
```

```
graph = [Edge(0, 1, 1), Edge(1, 2, 1), Edge(2, 0, 1),  
Edge(2, 3, 1), Edge(3, 1, 1)] # wagi
```

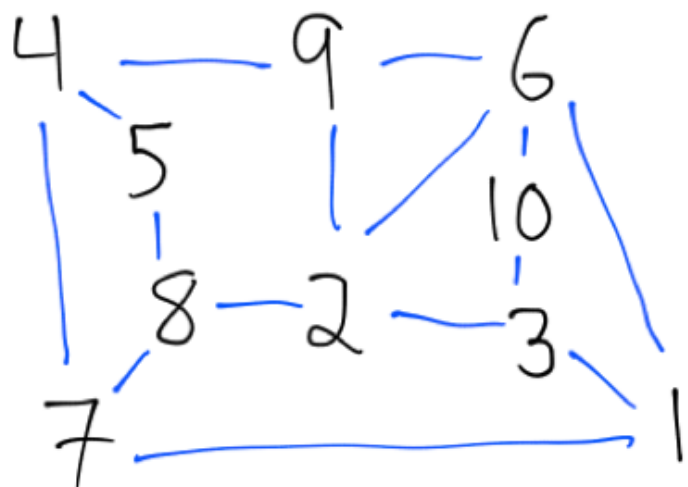
```
graph = [  
  [-1, 0, 1, 0, 0],  
  [1, -1, 0, 0, 1],  
  [0, 1, -1, -1, 0],  
  [0, 0, 0, 1, -1]] # bez wag
```


Reprezentacja macierzy sąsiedztwa

Graf $G=(V,E)$

- V – zbiór wierzchołków,
- E – zbiór krawędzi

Relacją sąsiedztwa (adiancja)



adjacency matrix

	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	0	1	1	0	0	0
2	0	0	1	0	0	1	0	1	1	0
3	1	1	0	0	0	0	0	0	0	1
4	0	0	0	0	1	0	1	0	1	0
5	0	0	0	1	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	1	1
7	1	0	0	1	0	0	0	1	0	0
8	0	1	0	0	1	0	1	0	0	0
9	0	1	0	1	0	1	0	0	0	0
10	0	0	1	0	0	1	0	0	0	0

Rozwiązanie specyficzne dla systemów

Reprezentacja graphml

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml ...>
<key id="d0"for="node" attr.name="color" attr.type="string"> <default>yellow</default>
</key>
<key id="d1" for="edge"attr.name="weight" attr.type="double" /> <graph id="G"edgedefault="undirected">
<node id="n0">
<data key="d0">green</data>
</node>
<node id="n1" />
<node id="n2">
<data key="d0">blue</data>
</node>
<edge id="e0" source="n0" target="n2"> <datakey="d1">1.0</data>
</edge>
<edge id="e1" source="n0" target="n1">
<data key="d1">1.0</data>
</edge>
<edge id="e2" source="n1"target="n3">
<data key="d1">2.0</data>
</edge>
</graph>
</graphml>
```

Za wykładem Szymon Matejczyk

Graph drawing – zasady budowy wizualizacji

Rysowanie grafów (lub sieci) – graficzna reprezentacja węzłów i krawędzi:

- Dany graf (reprezentowany matematycznie) – może być wizualizowany na wiele sposobów
- Ułożenie (pozycja na rysunku) węzłów i krawędzi na medium oraz wykorzystanie innych środków wizualizacji – poprawiać zrozumienie struktury przez odbiorcy, być użyteczne i ułatwiać eksplorację, analizę

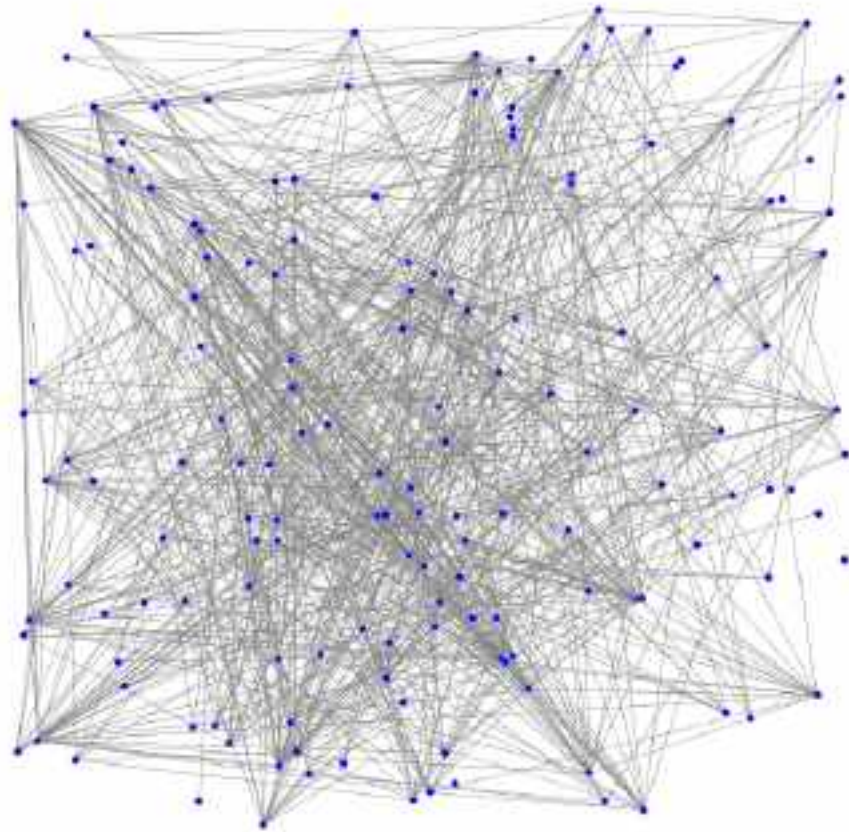
Historyczne zasady rysowania grafów

Node-link diagrams – Ramon Llull (XIIIw.)

Algorytmizacja – **Layout algorithms**

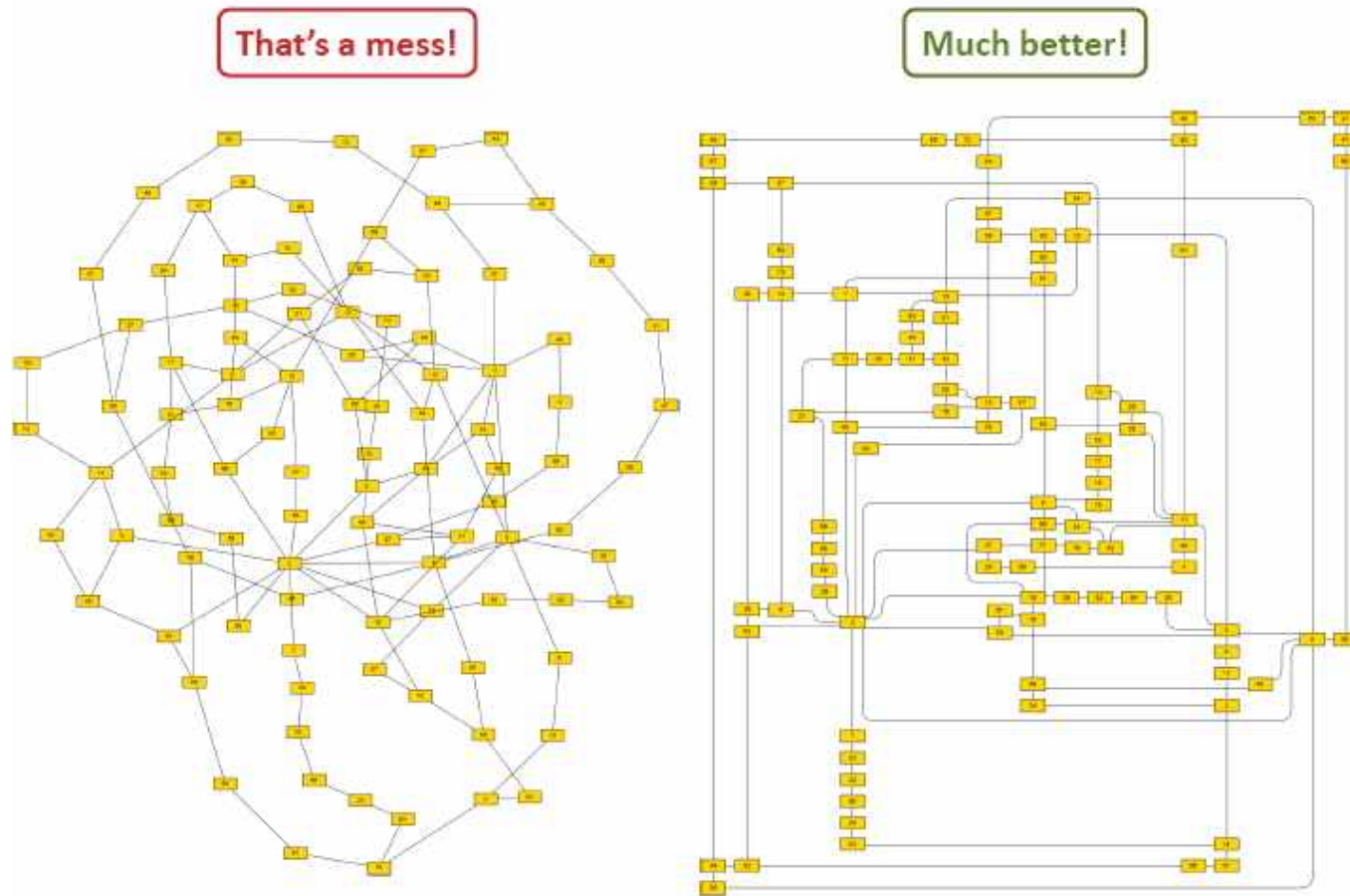
Przegląd – G. Di Battista et al. Graph drawing: Algorithms for the visualization of graphs.

Wizualizacja grafów powinna być czytelna



Nie jest to trywialne zadanie

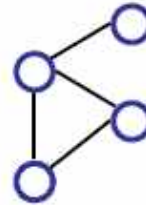
Jak rysować grafy?



Zasady tworzenie wizualizacji grafów

Aesthetic criteria

- minimize edge crossings

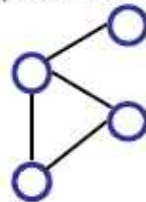


better than

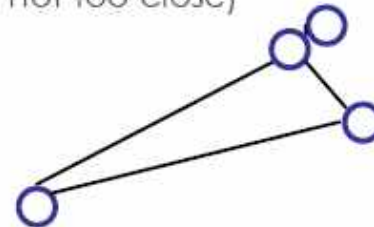


- uniform edge lengths

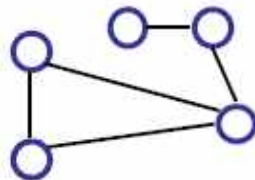
- (connected nodes close together but not too close)



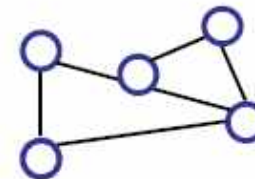
better than



- don't allow nodes to overlap with edges that are not incident on them



better than



Graph drawing - inne miary oceny wyniki

Stopień przecięcia:

- Względna liczba przecinających się krawędzi

Symetria (zwłaszcza dla drzew)

Prostota

Minimalizacja długości krawędzi

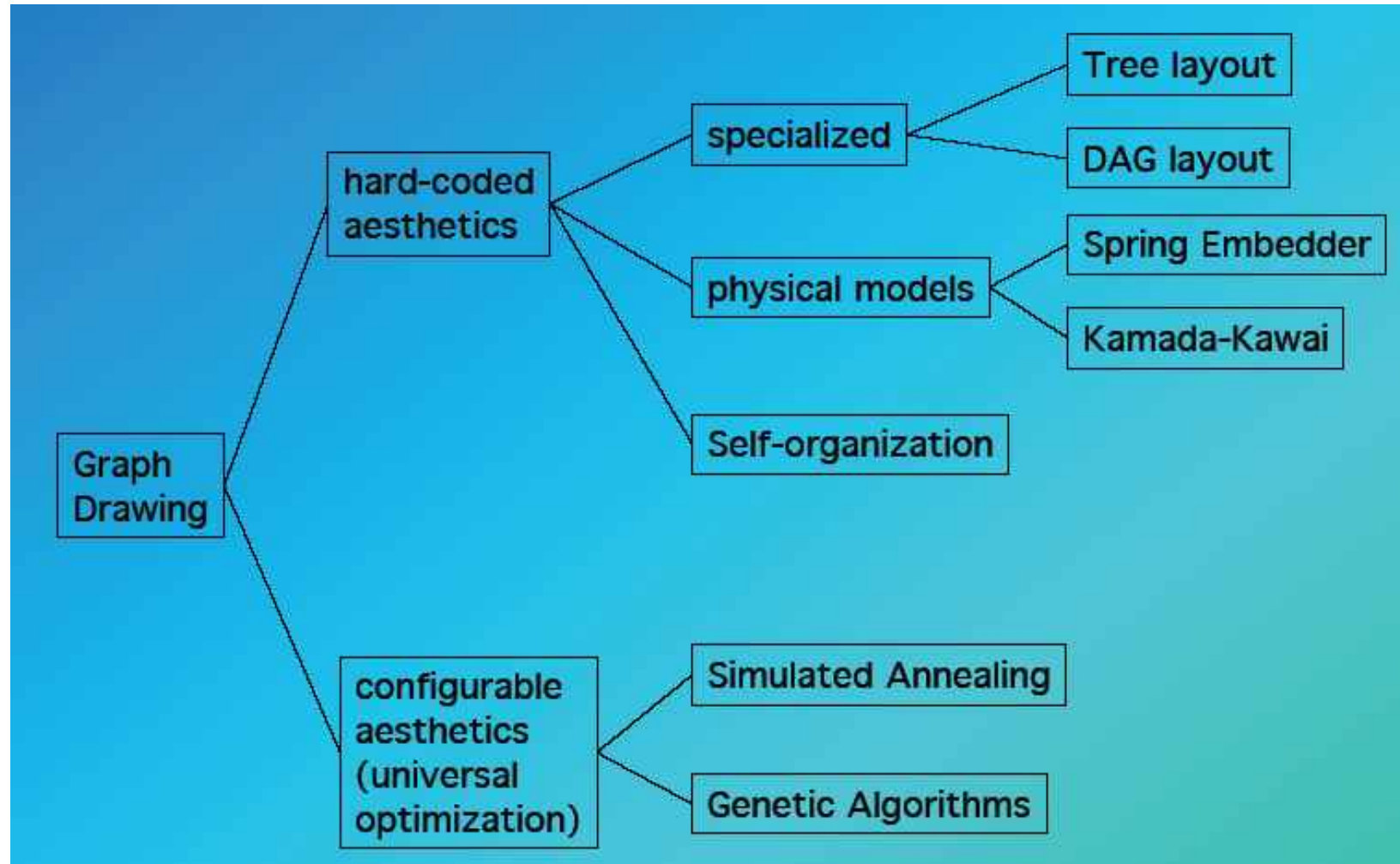
Gęstość rozłożenia elementów na powierzchni

Dobór rozdzielczości (kątovej)

- Także analiza liczby wierzchołków w podgrafach

Przegląd – G. Di Battista et al. Graph drawing: Algorithms for the visualization of graphs.

Klasyfikacja graph layout methods



Podstawowe algorytmy rysowania grafów

Algorytmy zbudowane z wykorzystaniem metafory fizykalnych systemów (wywodzącą się od Tutte [1963]):

- Siły sprężystości połączeń – ang. **force spring algorithm**
- Zasady mechaniki molekularnej

Algorytm kołowy (radial based) – wymaga przetworzenia grafu spójnego do postaci drzewa

Inne specjalizowane rozwiązania dla struktur drzewiastych

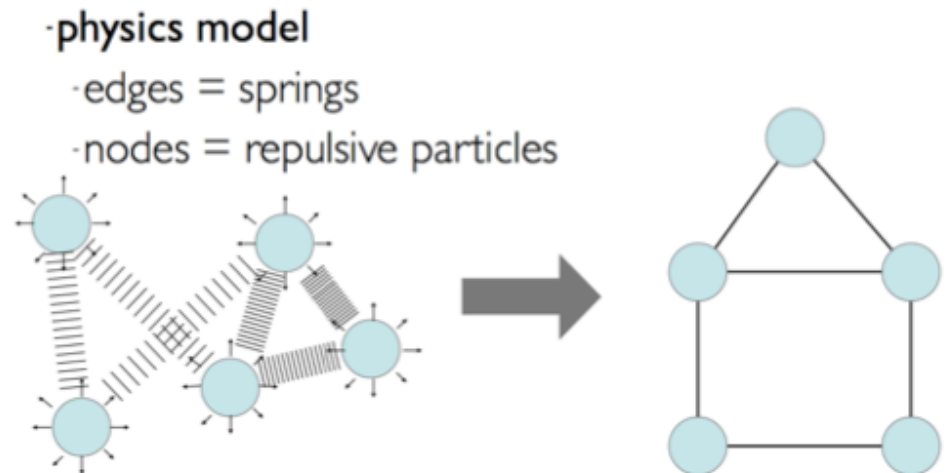
Algorytm symulacji sił fizycznych

Fruchterman and Reingold, 1991, Graph Drawing By Force-Directed Placement

Algorytm z wykorzystaniem metafory modelu sił fizycznych oddziałujących na wierzchołki (ang. force directed)

Wierzchołki oddziałują na siebie siłą odwrotnie proporcjonalną do odległości między nimi (jak ładunki elektryczne o tym samym znaku) tzw. the repulsive force – rozważa się wszystkie pary wierzchołków

Krawędzie – jak sprężyny, które przyciągają połączone wierzchołki proporcjonalnie do odległości (tzw. attractive forces)



FORCE MODEL

- many variations, but usually physical analogy:

- repulsion** : $f_R(d) = C_R * m_1 * m_2 / d^2$

- m_1, m_2 are node masses

- d is distance between nodes

- attraction** : $f_A(d) = C_A * (d - L)$

- L is the rest length of the spring

- i.e. Hooke's Law

- total force on a node x with position x'**

- $\sum_{\text{neighbors}(x)} : f_A(||x'-y'||) * (x'-y') + -f_R(||x'-y'||) * (x'-y')$

Layout algorithm – force based strategies

Rozpocznij od wstępnego (losowego) ułożenia obiektów

Spring forces

- Zamodeluj siły przyciągania pomiędzy sąsiednimi węzłami
- Uwzględnij odpychające siły wszystkich par węzłów
- Działaj krokowo dla minimalizacji energii systemu poprzez krokowe przesunięcia węzłów oraz zmian sił pomiędzy węzłami
- Cel – minimalizacji długości krawędzi oraz dobra separacja położenia węzłów

Matematycznie – odmiana spadku gradientu minimalizującego pewną formę energii układu albo modelu sił przesuwających się wierzchołków

Zmiany w ograniczonym obszarze wokół wierzchołków (koszty obl)

Wykonywany iteracyjnie z elementami tzw. symulowanego wyżarzania

Komentarz do schematu postępowania

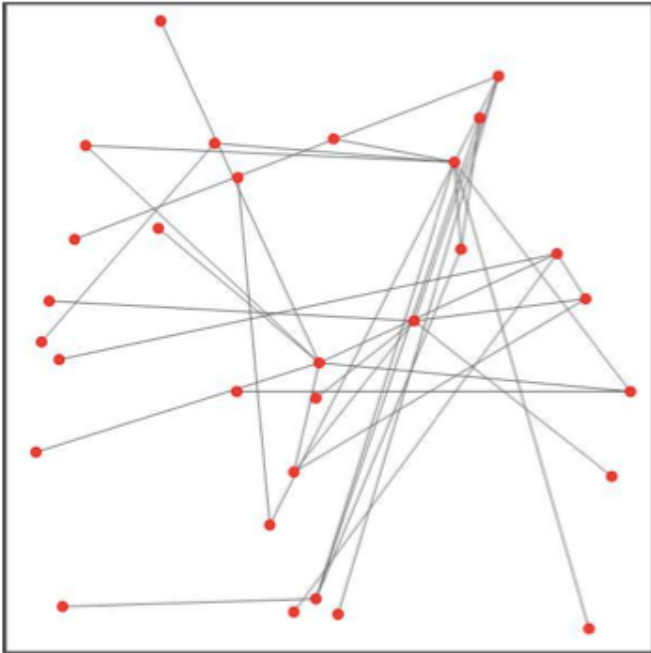
Obliczana jest powierzchnia obszaru do rysowania.

Zmienna (temperatura) inicjalizowana jest jedną dziesiątą długości dłuższego boku powierzchni. Obliczana jest wartość współczynnika k , który ogranicza przesunięcie.

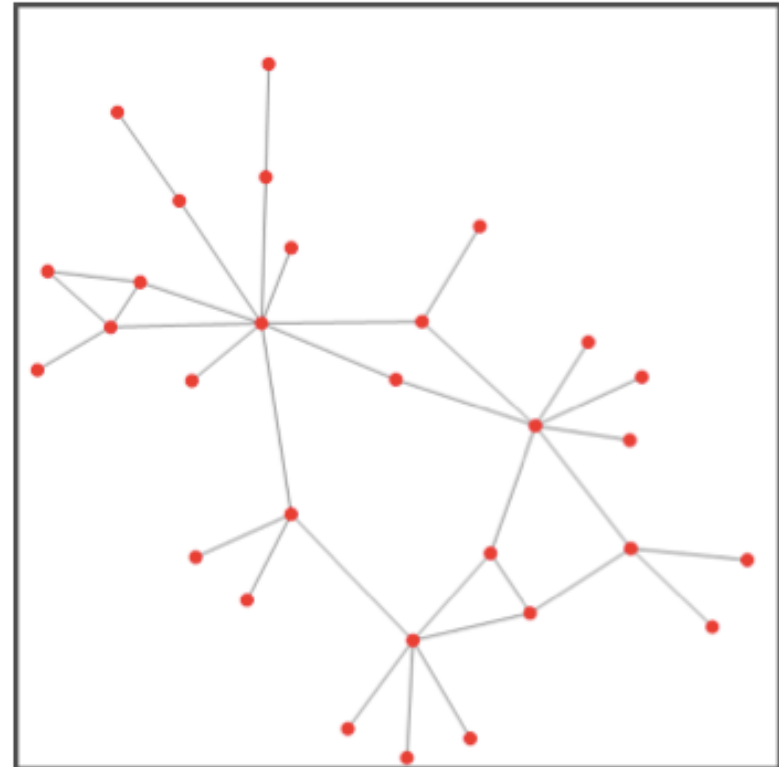
Następnie w n iteracjach wykonywane są kolejno:

- Obliczenie wektorów przesunięć dla każdego wierzchołka w stosunku do pozostałych (siła odpychania).
- Obliczenie wektorów przesunięć dla każdej pary wierzchołków połączonych krawędzią (przyciąganie).
- Obliczenie nowych położeń wszystkich wierzchołków, z uwzględnieniem wektorów przesunięć. W tym kroku brana jest pod uwagę wartość temperatury (t).
- Zmniejszenie temperatury (chłodzenie).

```
procedure Layout(G = (V, E))
  initialize temperature t
  for i = 1 to n do
    for each v in V do
      disp(v) = 0
      for each u in V do
        if u ≠ v then
          Δ = pos(v) - pos(u)
          disp(v) = disp(v) + Δ/|Δ| · f_rep(u, v)
    for each (u, v) in E do
      Δ = pos(v) - pos(u)
      disp(v) = disp(v) - Δ/|Δ| · f_attr(u, v)
      disp(u) = disp(u) + Δ/|Δ| · f_attr(u, v)
    for each v in V do
      pos(v) = pos(v) + disp(v)/|disp(v)| · min(|disp(v)|, t)
    t = cool(t)
```



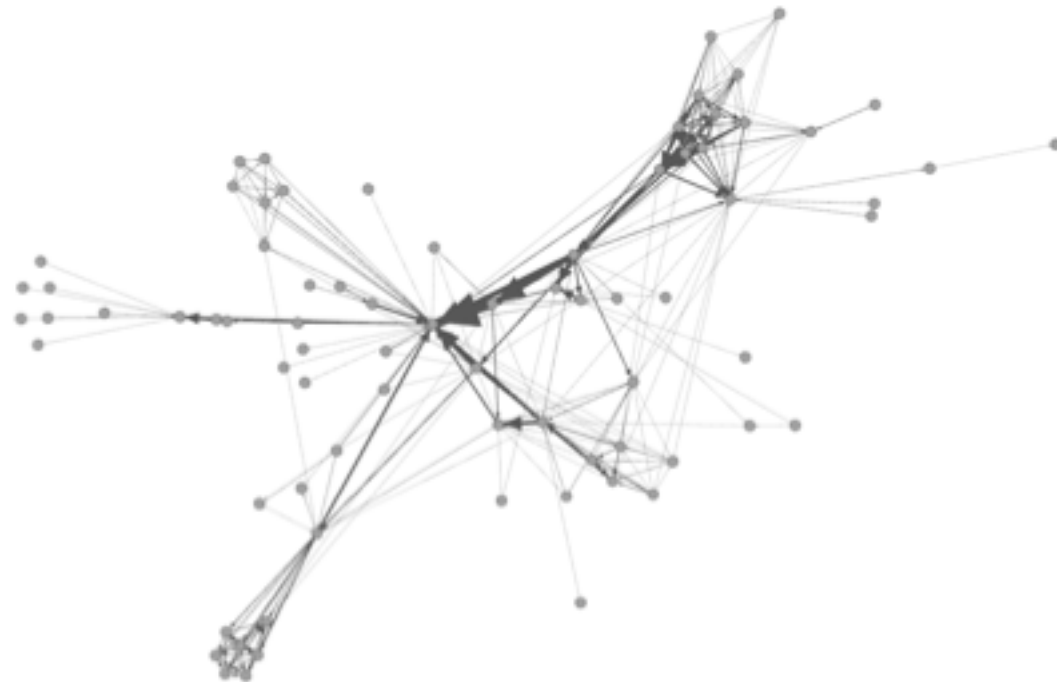
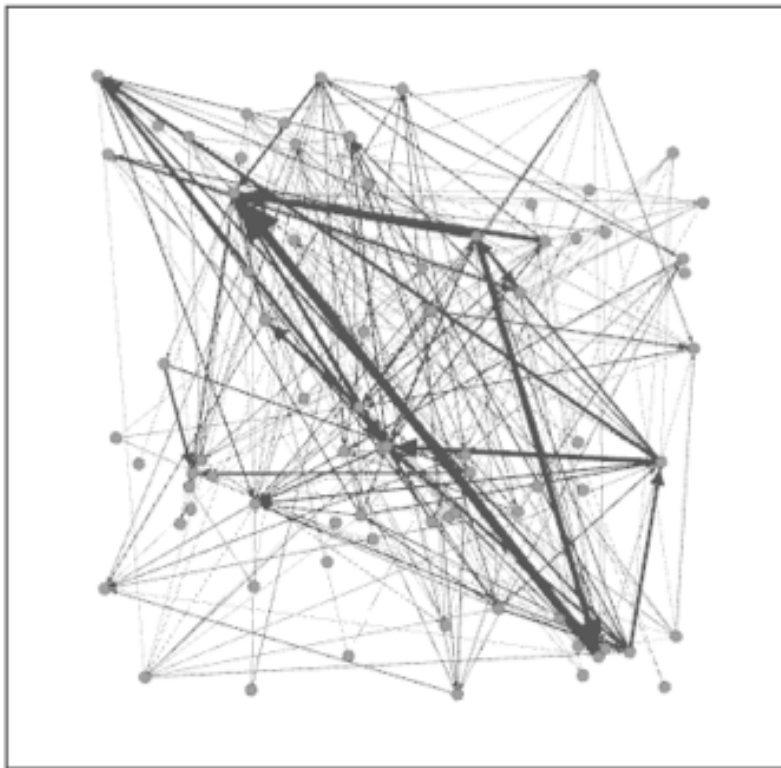
Przykładowy graf o losowym położeniu wierzchołków



Graf ułożony algorytmem Fruchtermana-Reingolda

- Za art. P.Pilny, A.Piłat

- Przykład analizy danych o powiązaniach bohaterów książki W.Hugo “Nędznicy” (ważone siłą interakcji)



Rozszerzenie Kamada i Kawai

- Problem ułożenia węzłów grafu jako proces stopniowej redukcji całkowitej energii układu sprężyn łączących wierzchołki.
- Sprężyny to ścieżki – siły mogą się odpychać lub przyciągnąć w zależności od odległości
- Autorzy zrezygnowali z traktowania wierzchołków jako ładunków oddziałujących na siebie.
- Cyt: This function is defined in such a way that low energies correspond to layouts in which adjacent nodes are near some pre-specified distance from each other, and in which non-adjacent nodes are well-spaced [Tamassia].

Kamada-Kawai Method

Imagine that the n vertices in an on-screen graph are connected by **springs**. The graph becomes a dynamical system that tries to reach a minimum-energy state, where springs are, as much as possible, not stretched or compressed overmuch from their “relaxed-state length”.

$$Cost = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{2} k_{ij} (\| \mathbf{y}_i - \mathbf{y}_j \| - l_{ij})^2$$

$l_{ij} = L \cdot d_{ij}$

desired on-screen length of edge (i,j)
length of shortest path (n. of steps) from i to j
desired on-screen length of an edge

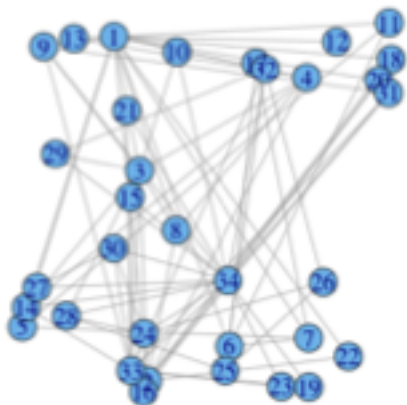
Use nonlinear optimization (gradient descent or the Newton-Raphson method) to minimize the energy with respect to the $\mathbf{y}_i$

$$k_{ij} = K / d_{ij}^2$$

constant strength of spring (i,j)

Gradients = Like the springs are pushing and pulling at the nodes

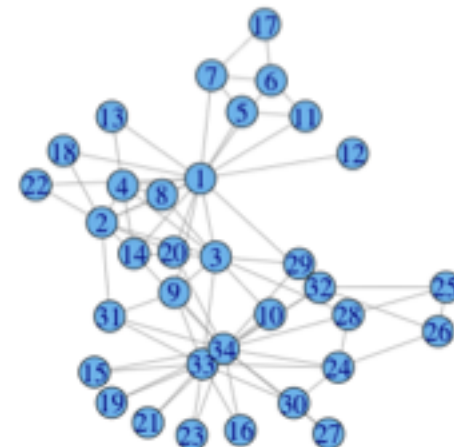
Przykłady ilustracyjne



Random layout



Kamada-Kawai



Fruchterman-Reingold

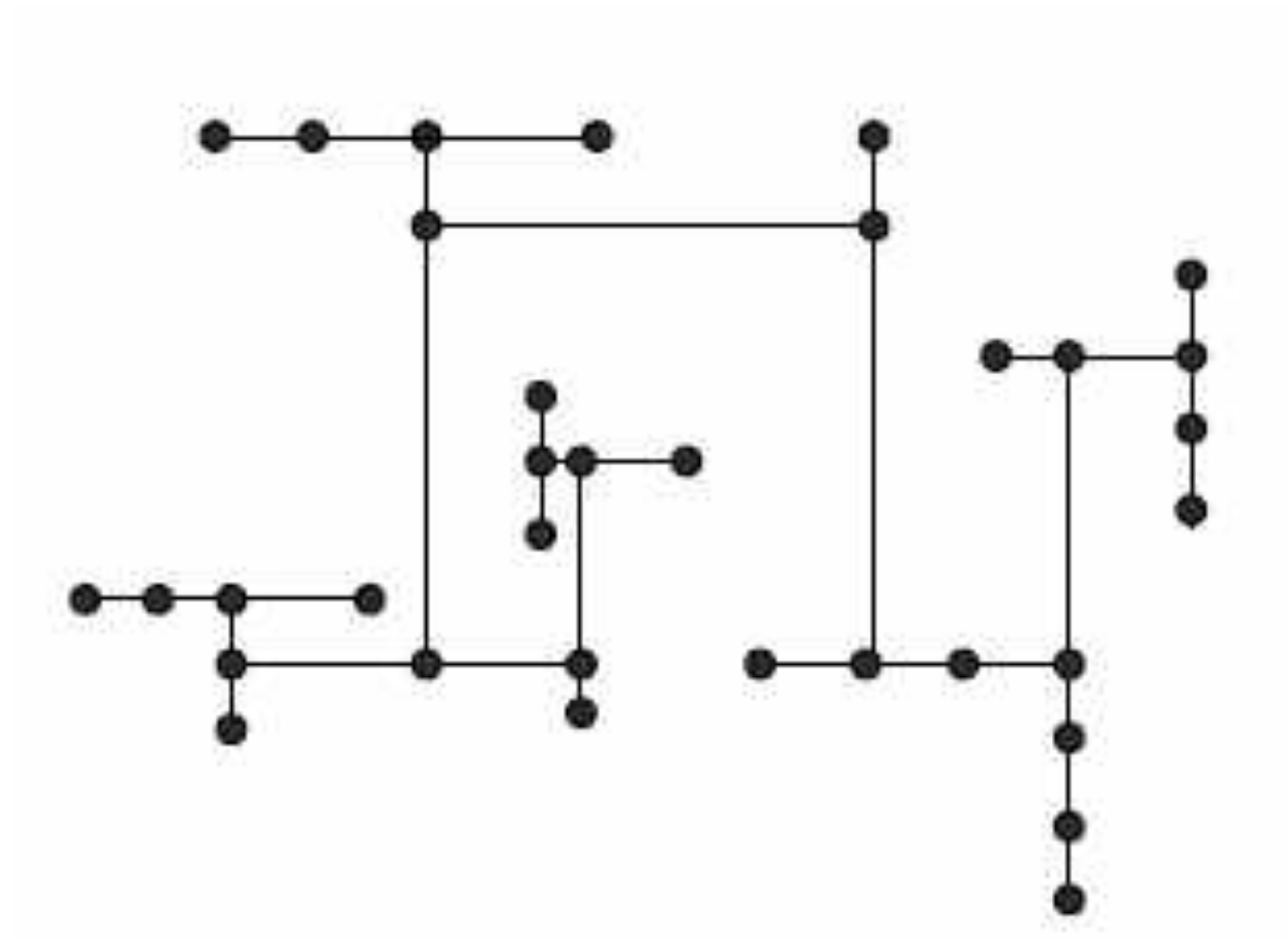
Przykłady R

-{igraph}: `layout.kamada.kawai()`, `layout.fruchterman.reingold()`

Uwagi nt. stosowania

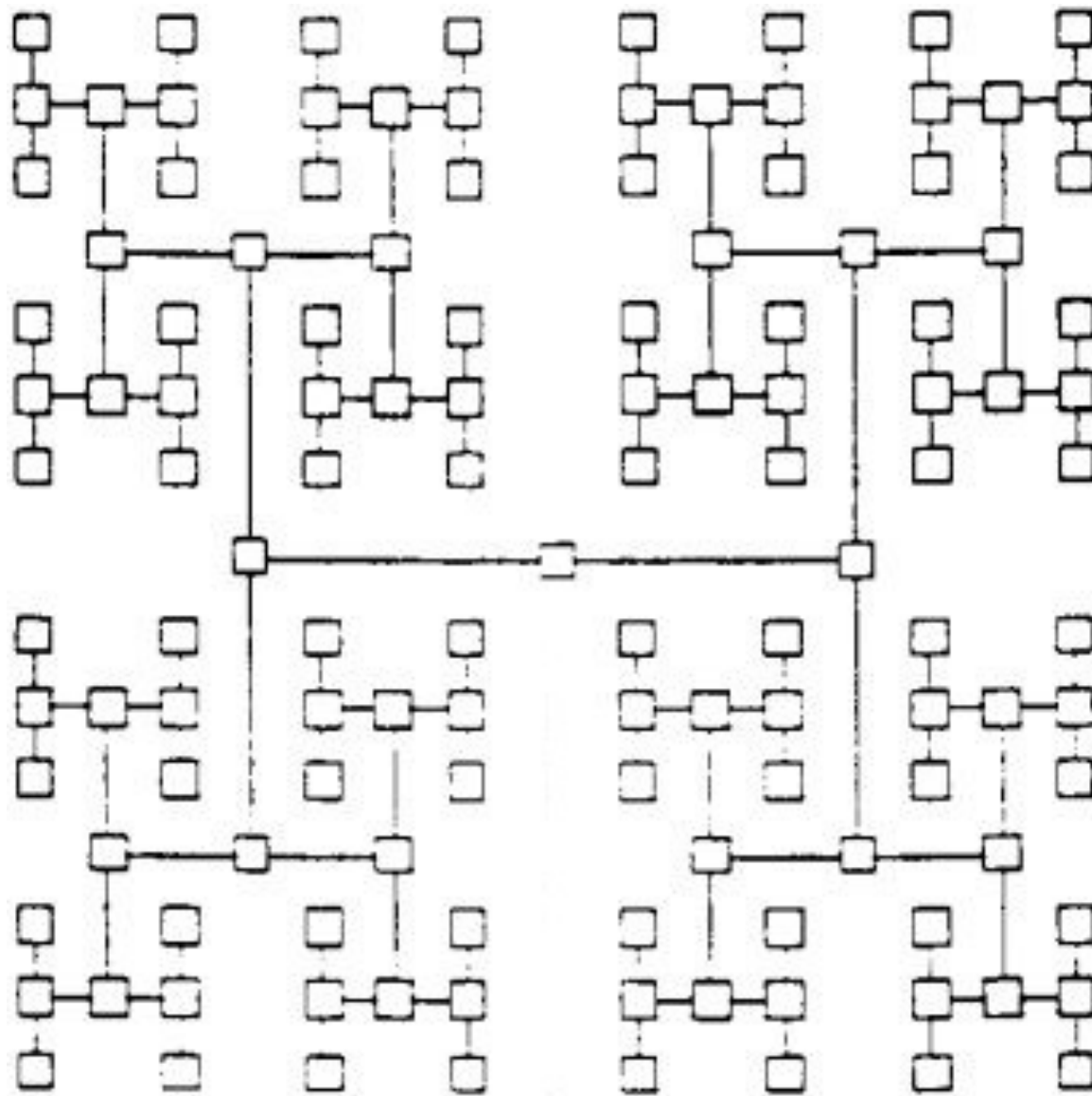
- Często zadawalająca jakość rysunku
 - Równomierne rozłożenie wierzchołków i dobre rezultaty nawet dla gęstych grafów
- Ograniczenie wobec rozmiarów grafów
- Duże koszty obliczeniowe –w najlepszym przypadku $O(n^3)$, gdzie n jest liczbą wierzchołków
- Przykład eskperymentu w tekście P.Pilny, A.Pelikant Wizualizacja i reorganizacja grafów
 - Graf 50 V i 73 E – 33 ms
 - Graf 100 V i 160 E – 125 ms
 - Graf 500 V i 791 E – 2840 ms
 - Graf 1000 V i 1555 E – 11 310 ms

H layered graph organization



Związki z projektowaniem VLSI

H layout



OTHER LAYOUTS

- **orthogonal**

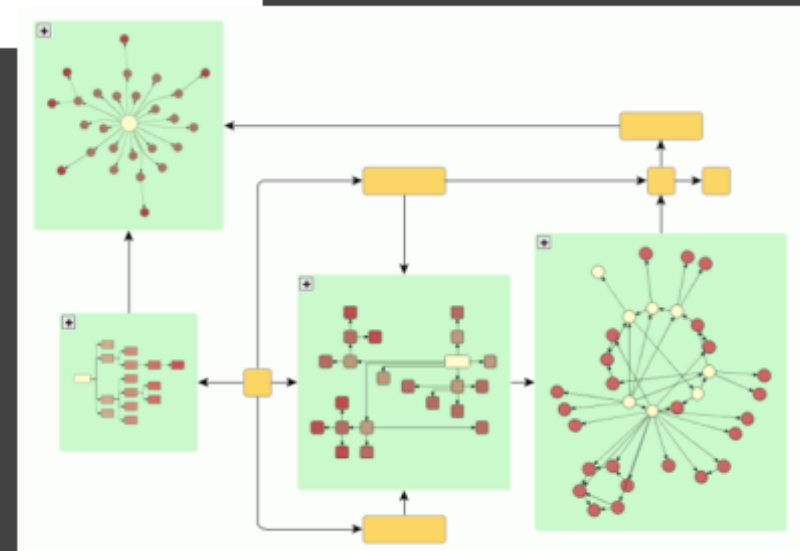
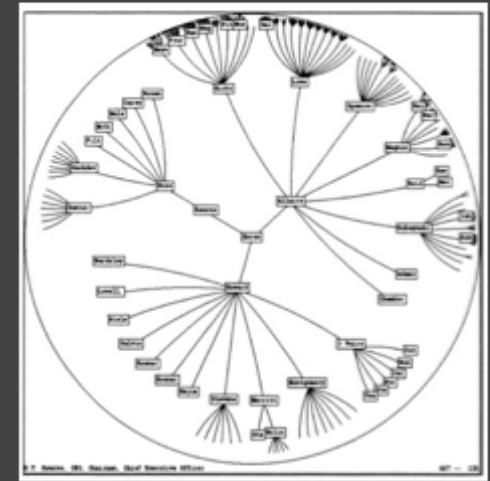
- great for UML diagrams
- algorithmically complex

- **circular layouts**

- emphasizes ring topologies
- used in social network diagrams

- **nested layouts**

- recursively apply layout algorithms
- great for graphs with hierarchical structure



Circular layout = algorytmy kołowe

Umieszczają wierzchołki na okręgu

Biorąc pod uwagę:

- Sąsiedztwo wierzchołków
- Redukcję potencjalnych przecięć krawędzi

Krawędzie mogą być różnie rysowane

Przykład algorytmu:

Six and Tollis 1999, A Framework for Circular Drawings of Networks



Circos

computes
circular layouts



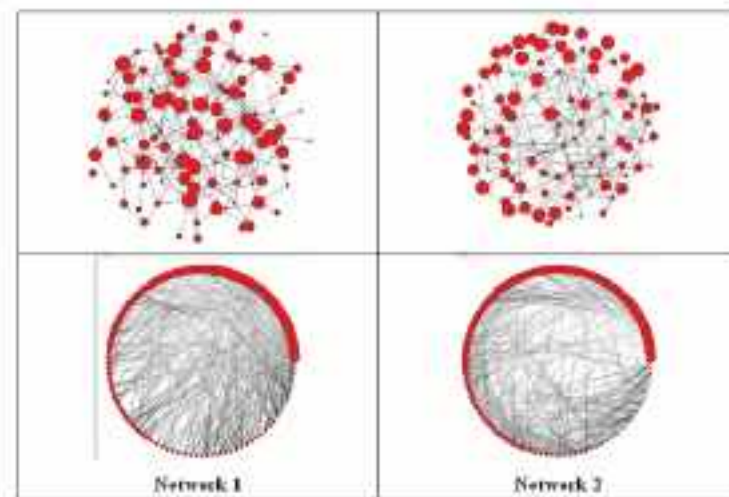
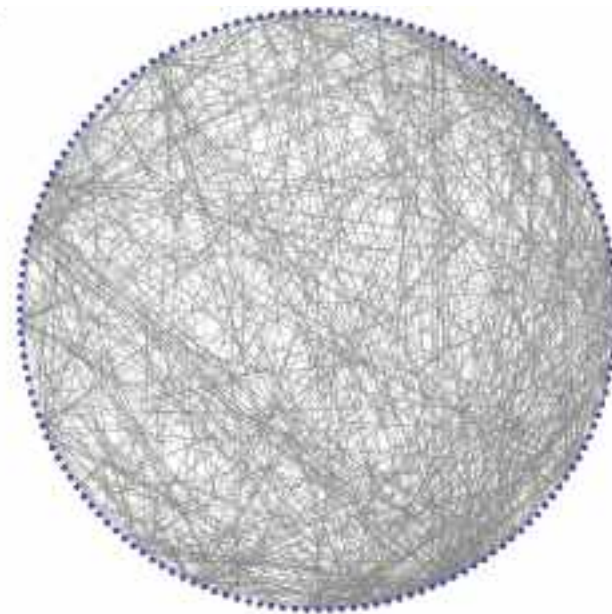
<http://circos.ca/>

Circular layout methods

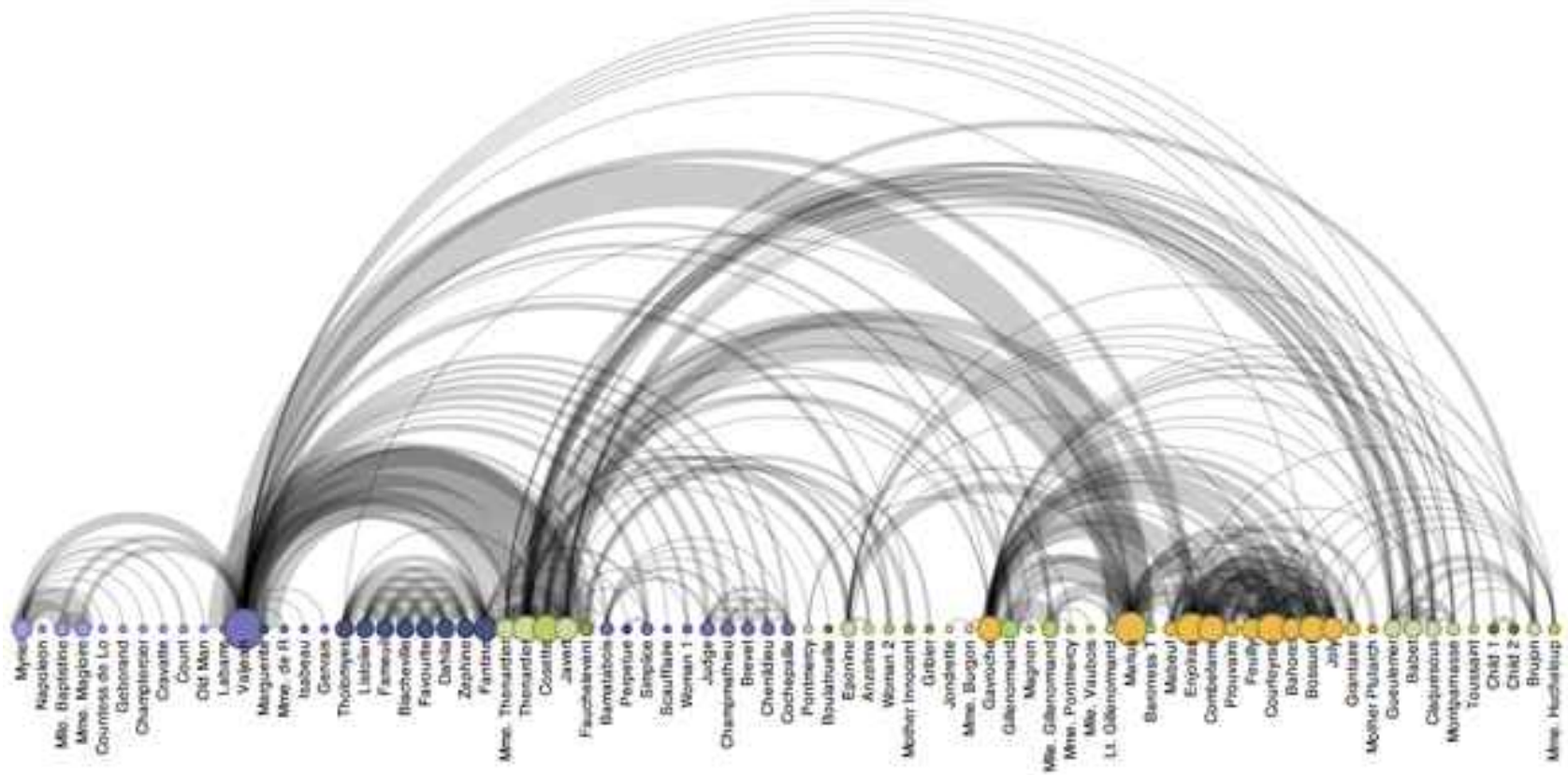
Krawędzie mogą być różnie rysowane

Trudności analizy:

- Zbyt dużych grafów
- Zbyt gęstych grafów



Arc graphs layout



Radial layout – algorytm radialny



Grant's Drawing of a Target Sociogram of a First Grade Class (from Northway, 1952).



McKenzie's Target Sociogram Board (fr Northway, 1952).

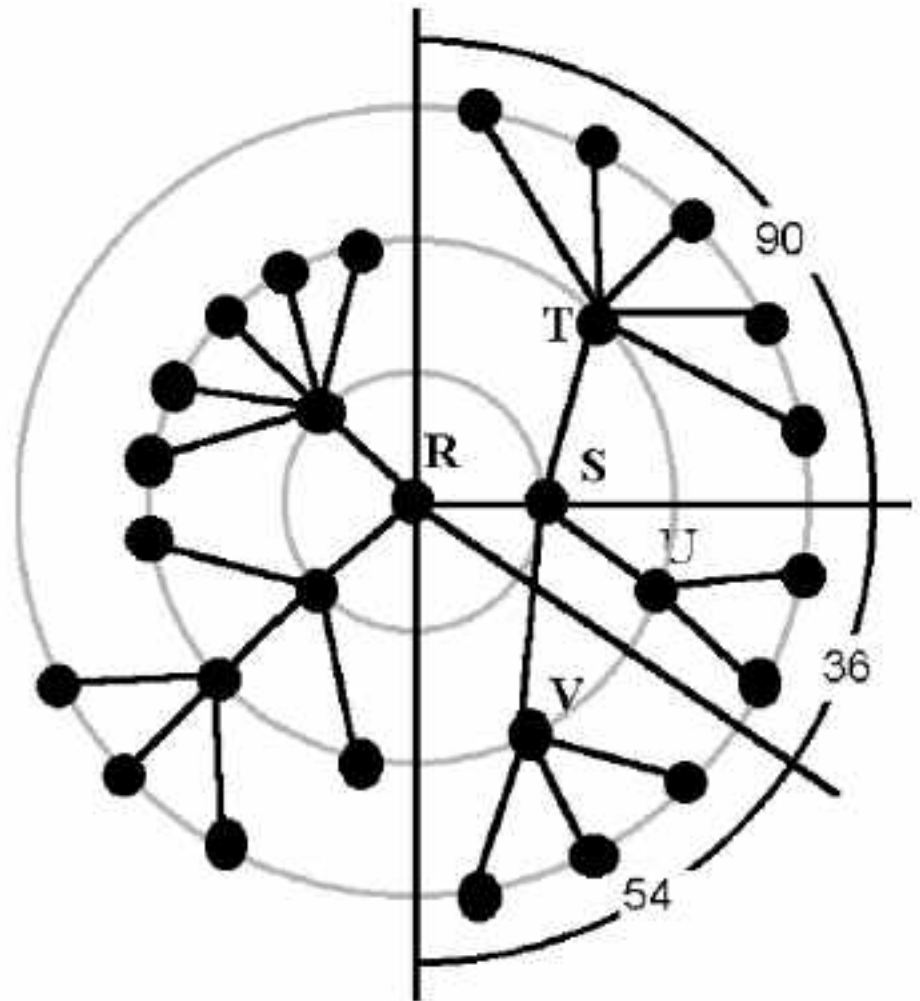
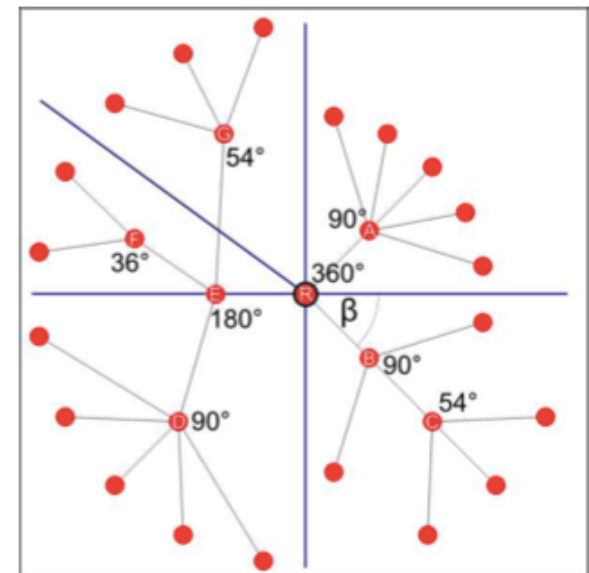


Fig. 8.1. *Radial placement.*

Grafy o strukturze, którą można zamienić do odmiany drzewa

Algorytmy radialne

- Wierzchołki pozycjonowane na koncentrycznych okręgach o promieniach zależnych od odległości w grafie danego wierzchołek od tzw. wierzchołka centralnego (należy przekształcić graf tak by była struktura drzewiasta – alg. BFS)
- Wierzchołek centralny pozycjonowany w środku i przypisuje się mu pełny kąt 360
- Wierzchołki najbliższe aktualnego wierzchołka centralnego – potomkowie – rozłożone wokół. Przypisuje się im wycinek kąta proporcjonalny do ilorazu liczby liści w jego poddrzewie.



Schemat podziału powierzchni dla układu radialnego

Radial layout

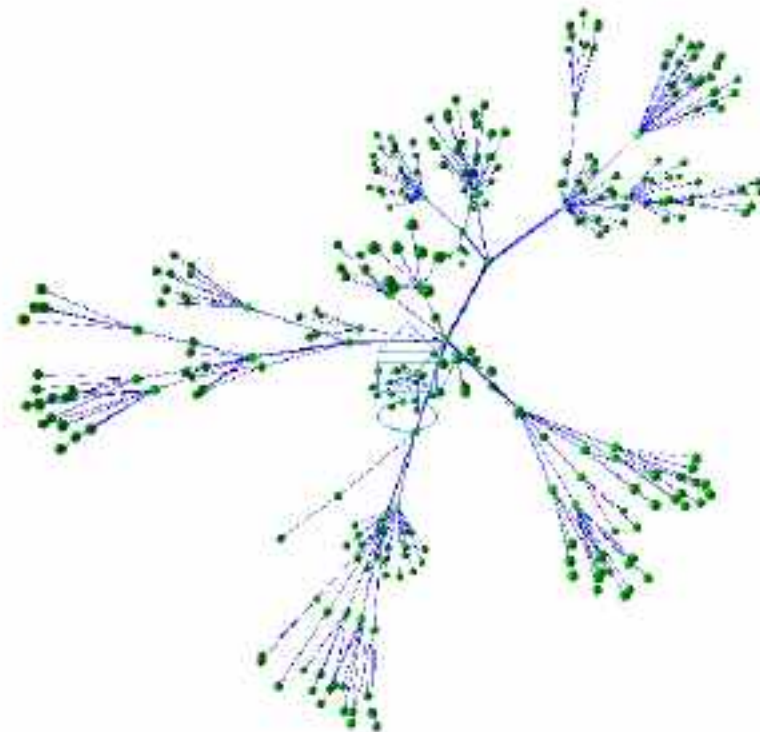
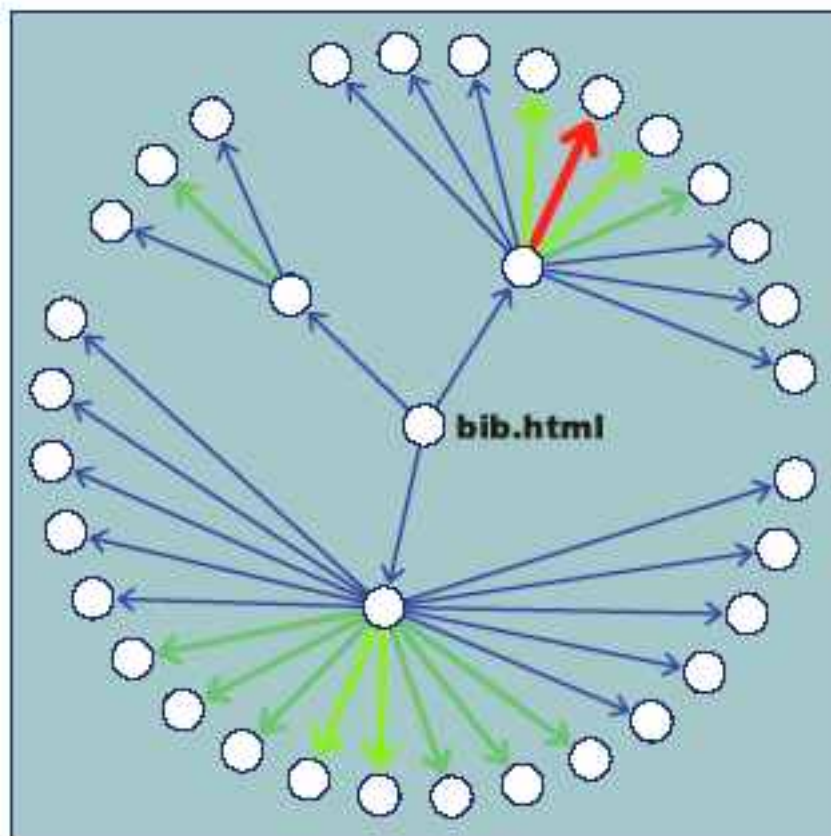
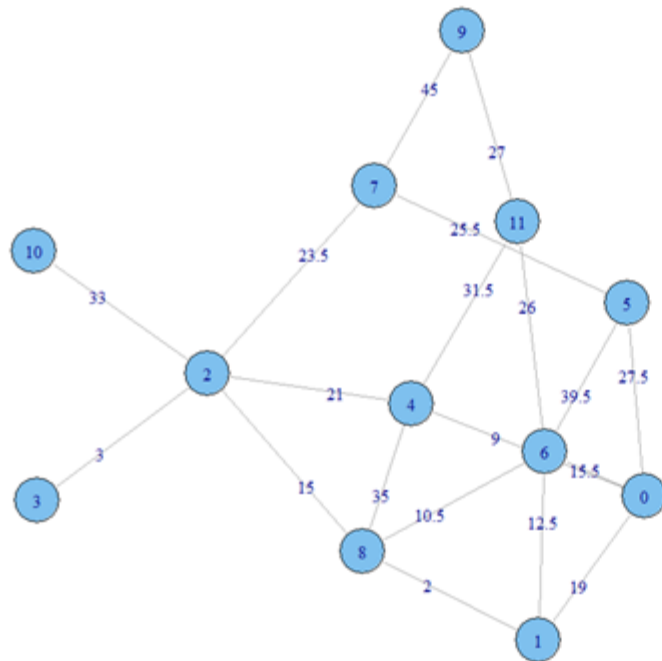


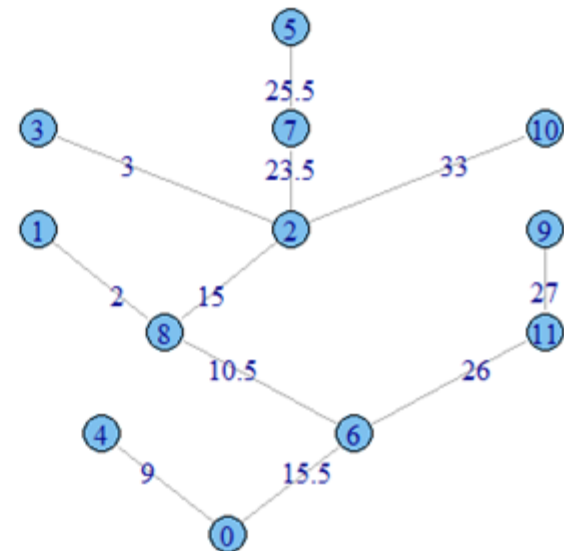
Fig 7. 3D version of a radial algorithm. (Courtesy of S. Benford, University of Nottingham, UK.)

Wersje 2D oraz b. zaawansowane 3D (z interakcją)

Minimalne drzewa rozpinające



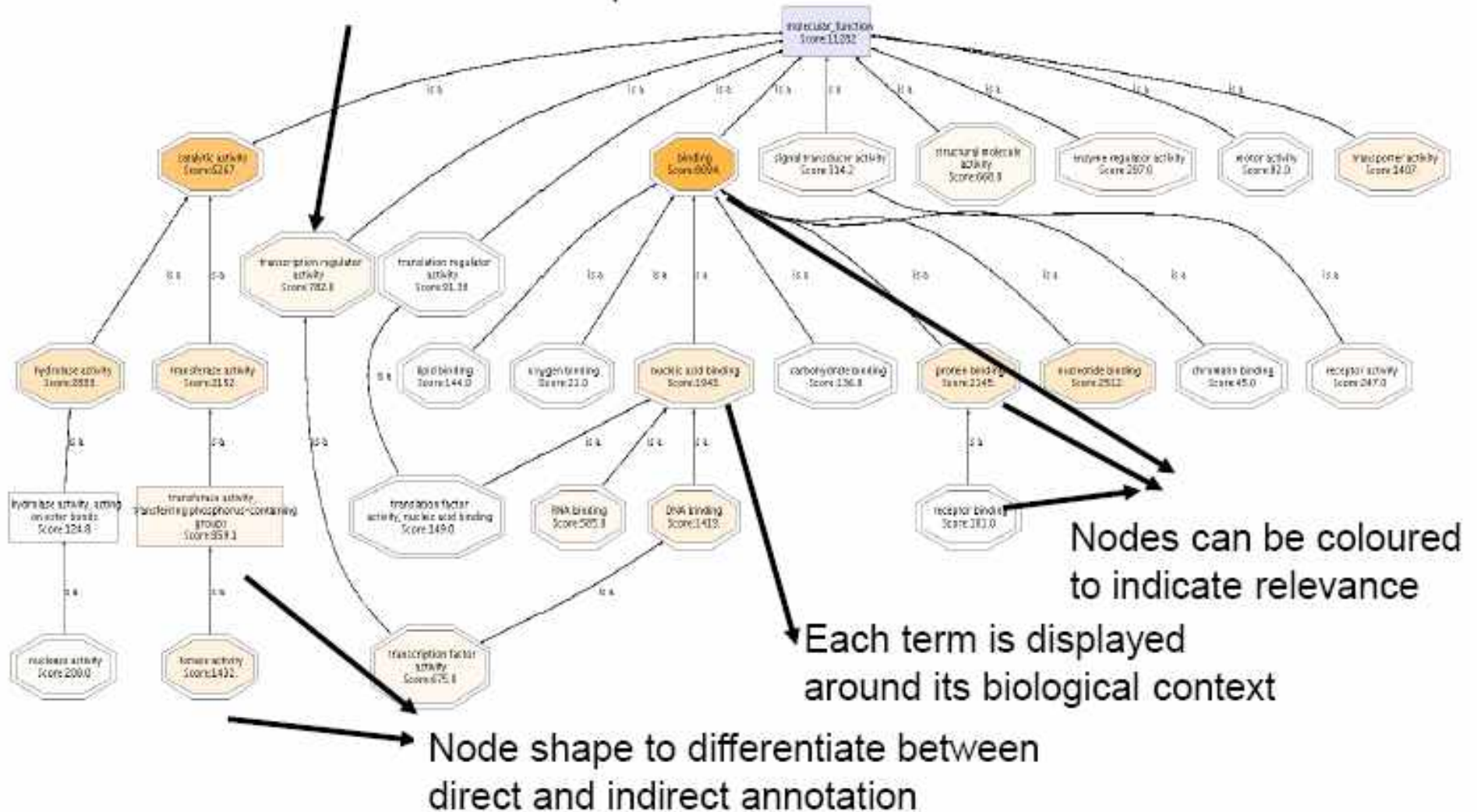
Original Graph



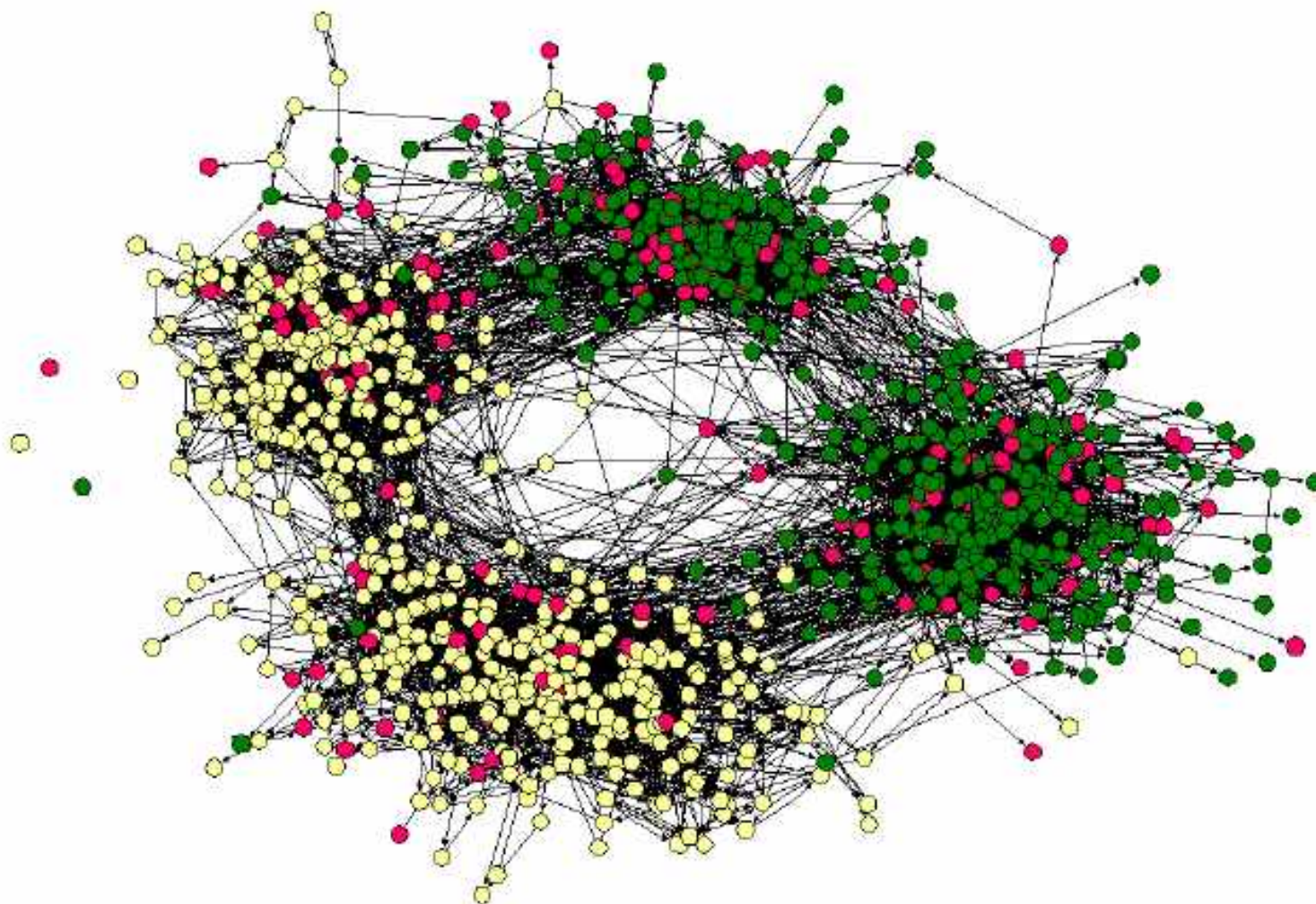
Minimum Spanning Tree

Dodatkowe pomysły na wyróżnianie

Each term has a number of sequences associated

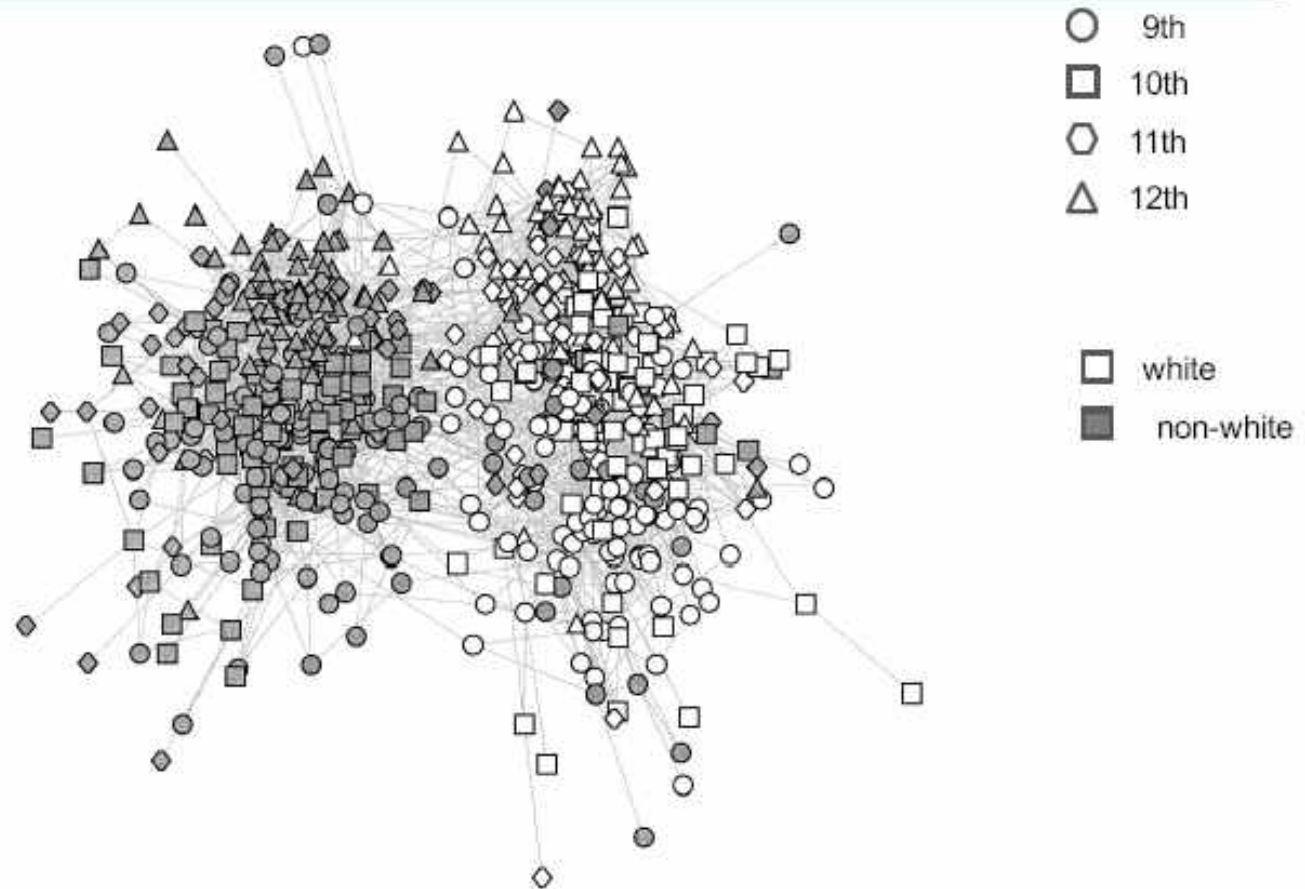


Case study – różne sposoby zwrócenia uwagi i eksploracji grafu



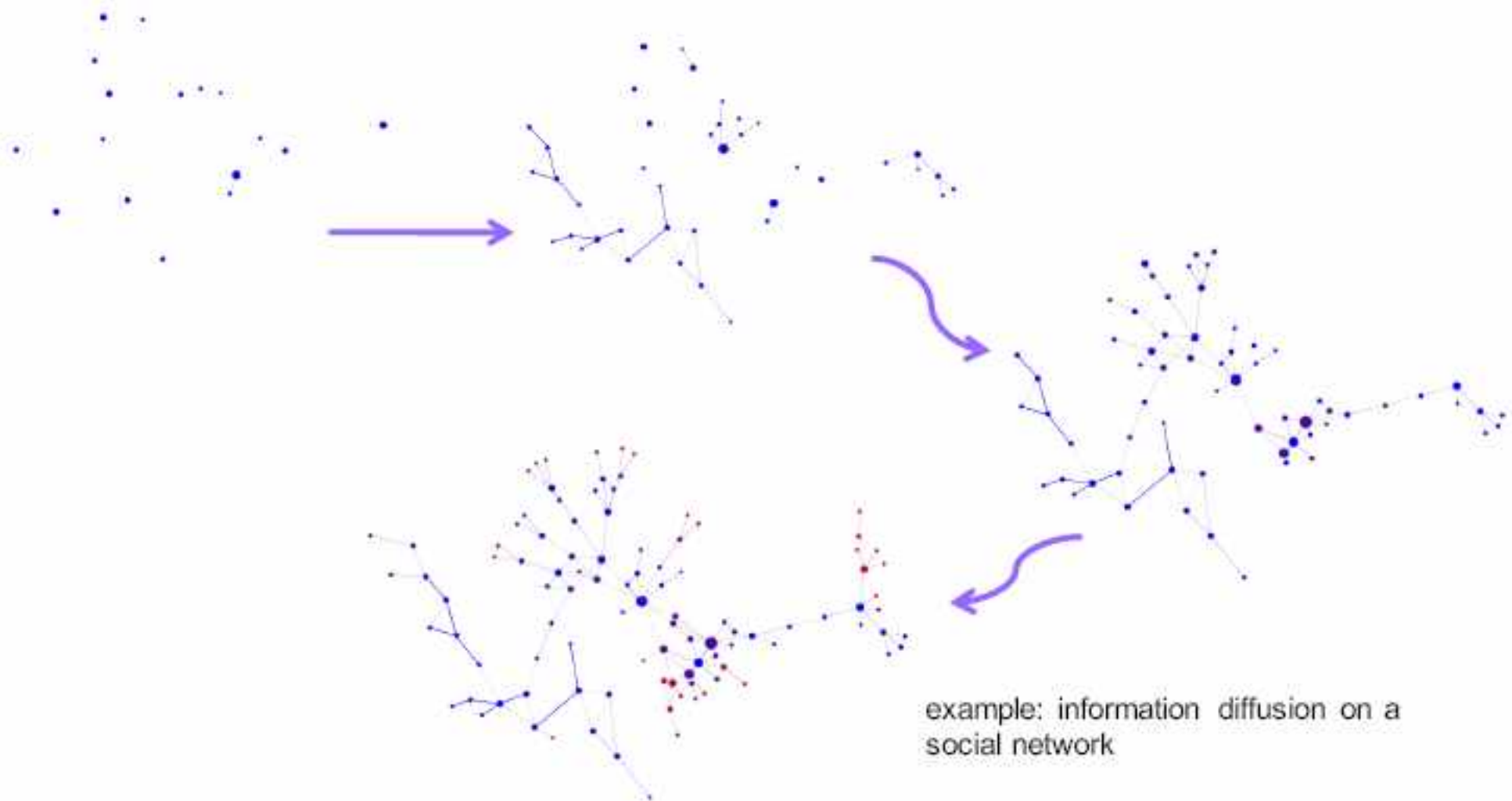
source: James Moody, **Race, School Integration, and Friendship Segregation in America**
AJS Volume 107 Number 3 (November 2001): 679–716

Case study – różne sposoby zwrócenia uwagi i eksploracji grafu



source: James Moody, **Race, School Integration, and Friendship Segregation in America**
AJS Volume 107 Number 3 (November 2001): 679–716

Zmienność grafu wraz z czasem – evolving data streams



On line graph drawing – data streams

- data abstraction: streaming data not static file
- task abstraction: dynamic stability (tradeoff)
 - minimize visual changes
 - stay true to current dataset structure



[Fig 1. Frishman and Tal. Online Dynamic Graph Drawing. Proc EuroVis 2007, p 75-82.]

Graph drawing – problem wielkości grafu

Zmiany wielkości grafów:

- Historycznie – grafy typowo kilkadziesiąt węzłów
- Duże grafy
 - Lata 90te – około 100 węzłów; a tysiące (huge graphs)
 - Nowe zastosowania – setki tysięcy i miliony wierzchołków

Poprzednio rozwinięte algorytmy rysowania grafów nieskuteczne

Nowe rozwiązania

Filter graphs, interaction, focus attention

Grupowanie

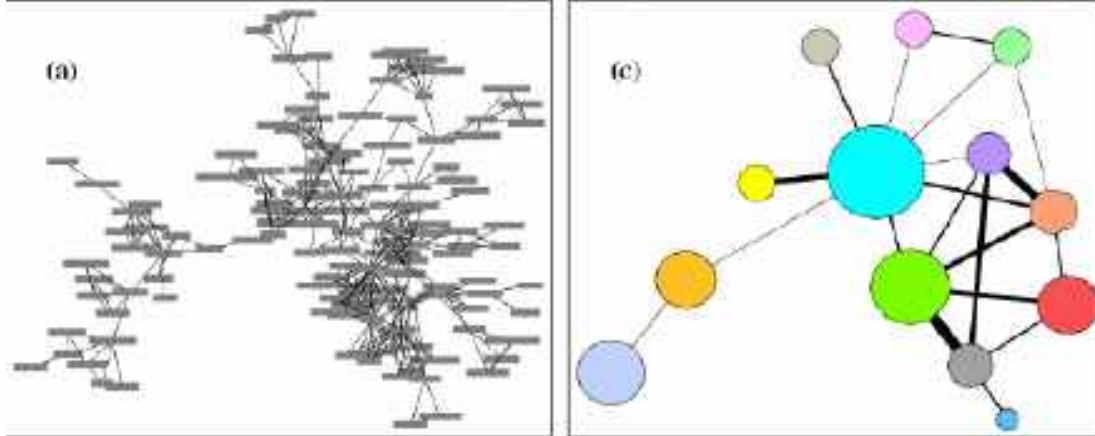
Próbkowanie oraz redukcja

Przegląd – Unwin A., Theus M., Hofmann: Graphics of Large Datasets

Typowe zadania eksploracji grafów

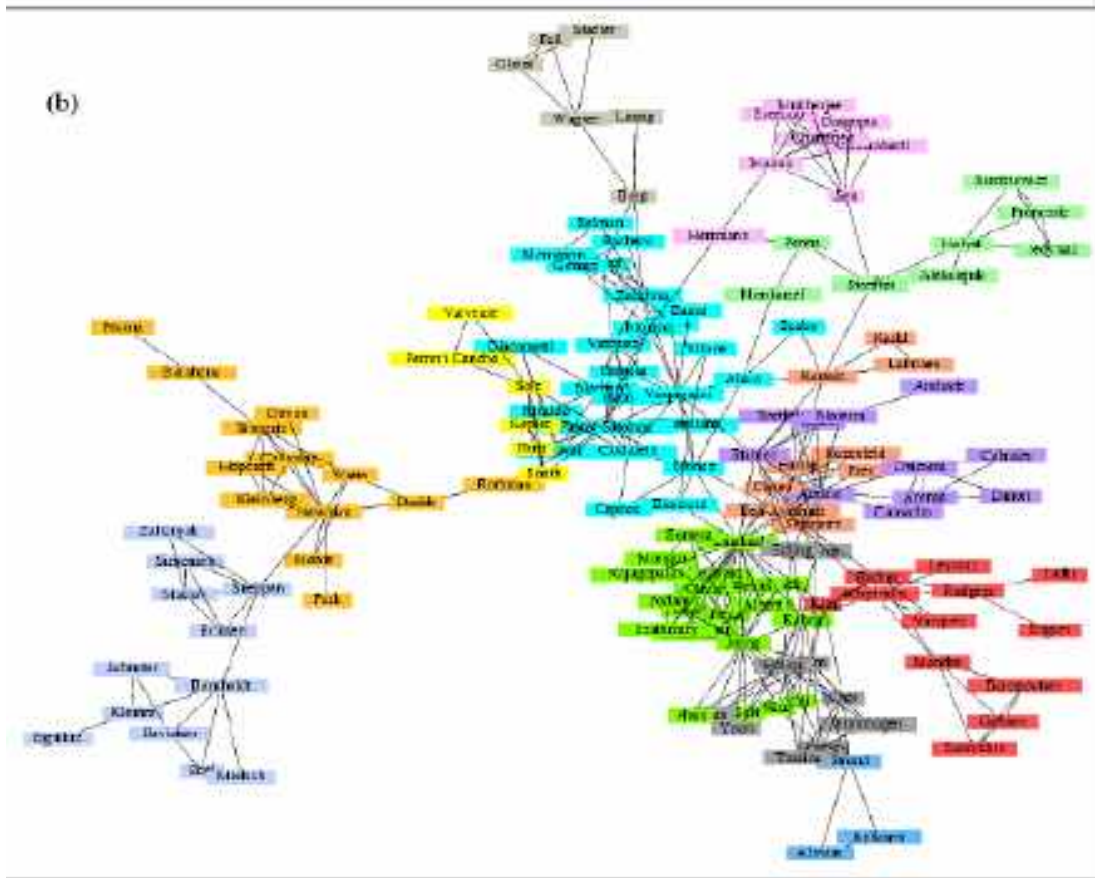
Tzw. Interactive Highlight techniques:

- Ułatwić zrozumienie struktury grafów
- Odkryć i pokazać nieznane właściwości
- Porównywać podgrafy (z uwagi na podobieństwo węzłów, krawędzi, informacji etykietującej).
- Poszukiwać podobieństw
- Odkrywać obserwacje odstające (outliers)
- Wykrywać podgrupy
- Umożliwiać tzw. „drill down” z ogólniejszego oglądu na podzbiory wyrażające określone zainteresowanie



Example of coarsening network structure

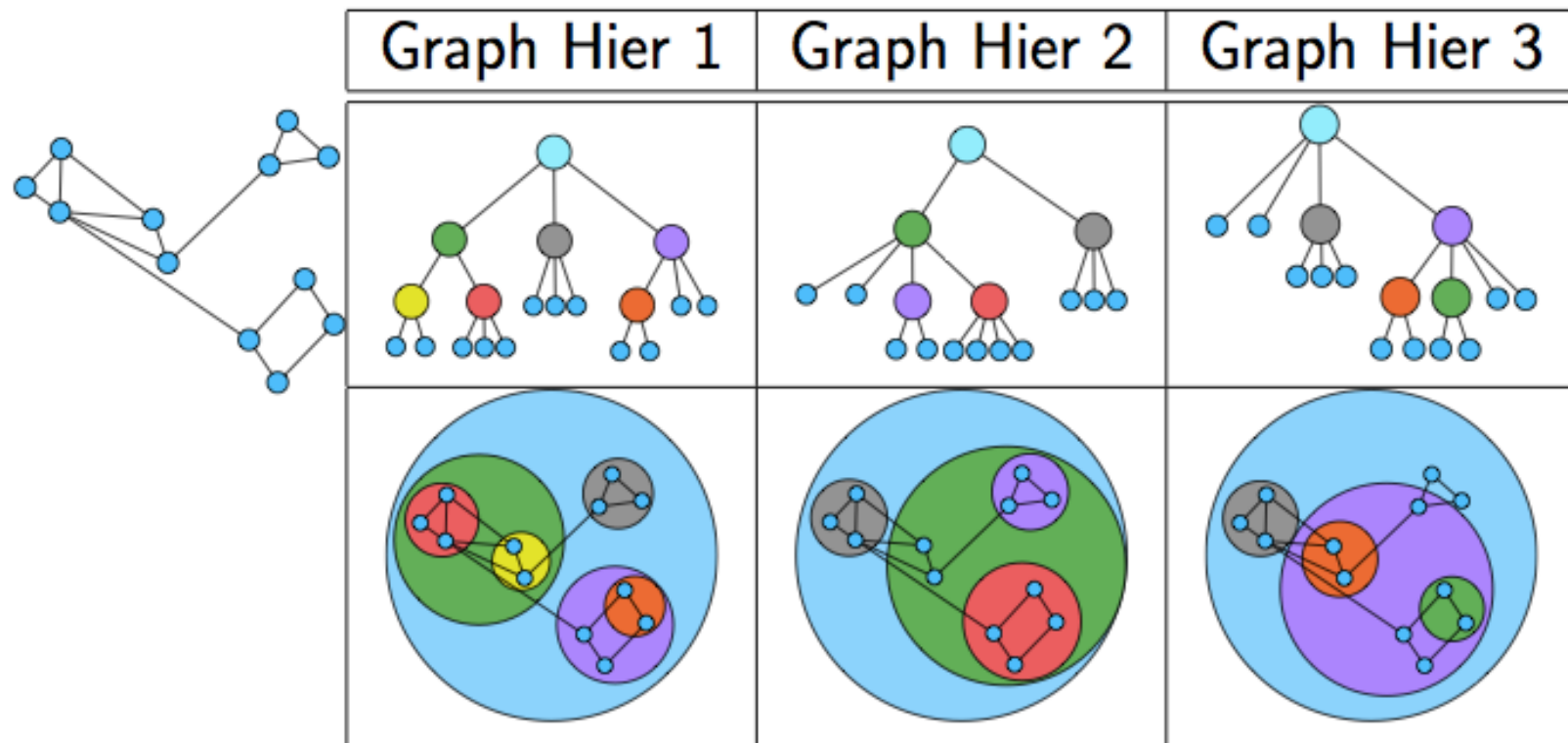
- Newman & Girvan 2004
- co-authorship network of physicists writing papers on networks
- clustering algorithm identifies different subcommunities
- each node is a community – size represents number of authors
- each edge thickness represents the number of co-author pairs between communities



Source: Finding and Evaluating Community Structure in Networks, M. E. J. Newman and M. Girvan,
<http://link.aps.org/doi/10.1103/PhysRevE.69.026113> DOI: 10.1103/PhysRevE.69.026113

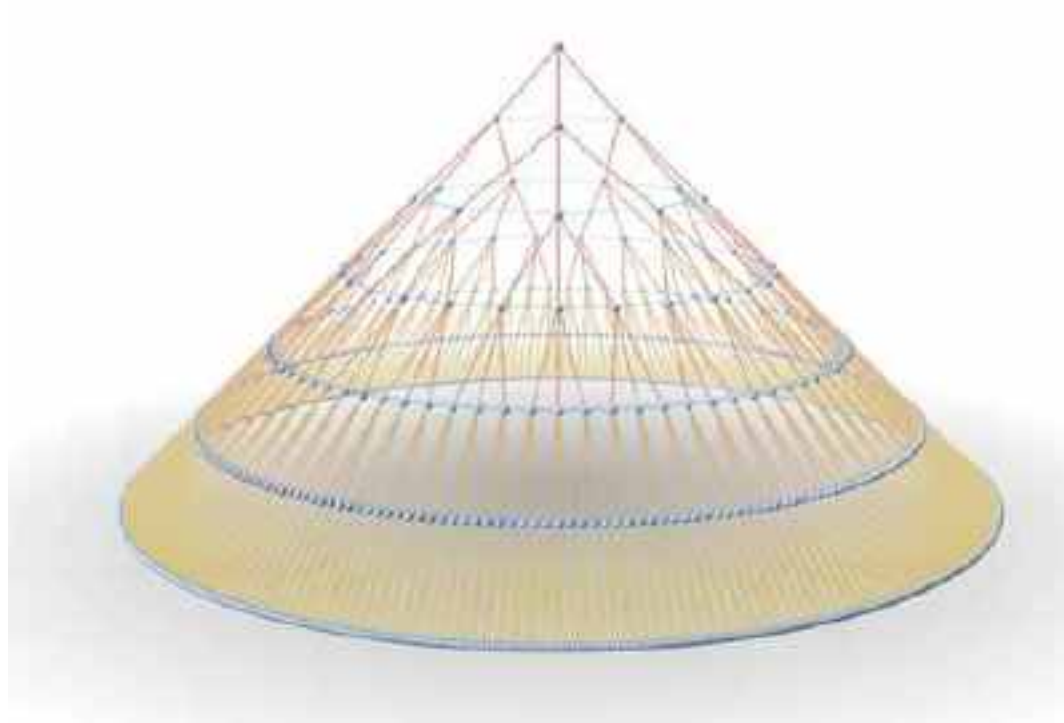
Wielo-poziomowość wizualizacji grafów

- data abstraction: create cluster hierarchy on top of original graph (coarsening)



[Fig 3. Archambault et al. GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. IEEE Trans. Visualization and Computer Graphics 14(4):900-913 2008.]

Wielo-poziomowość wizualizacji grafów



3 D cones trees

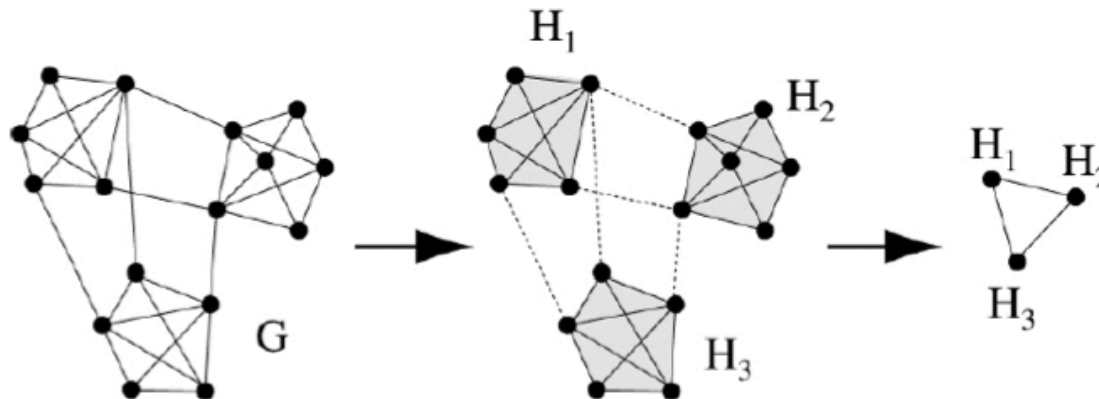
Transformacje grafów

Techniki grupowanie i upraszczanie wyglądu

- Slajd z wykładu Tamara Munzner: Information Visualization, UBC 2009

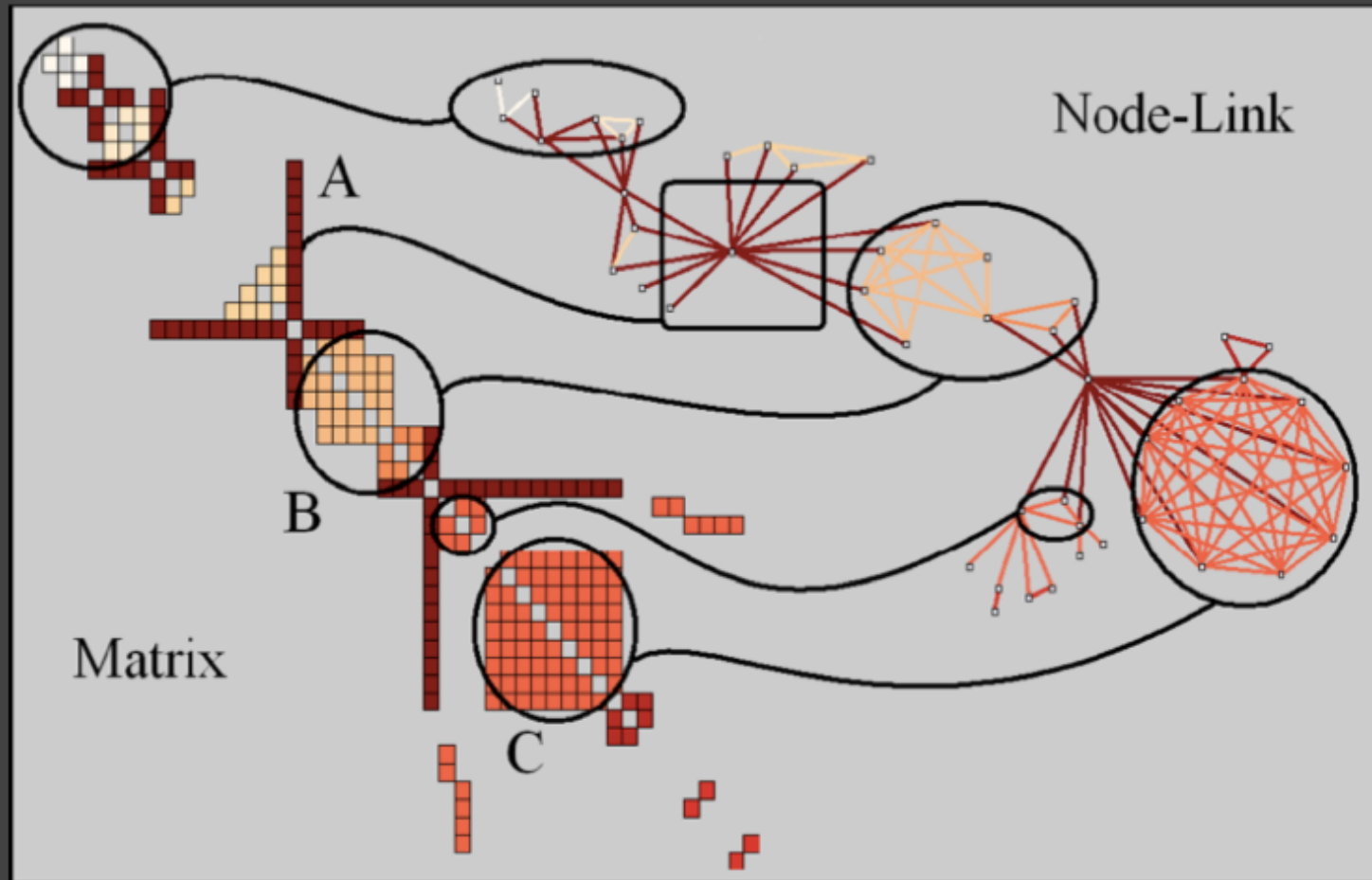
Small World Coarsening

- remove low-strength edges
- maximal disconnected subgraphs
- quotient graph: subgraph = higher-level node

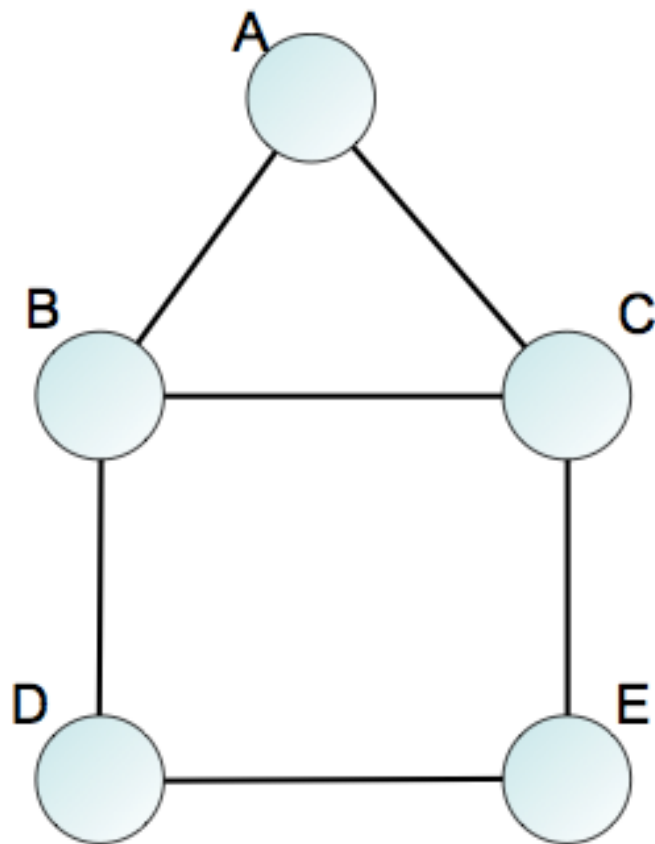


[Fig 2. Auber et al. Multiscale Visualization of Small World Networks. Proc. InfoVis 2003, p 75-81.]

Graph to także macierz



Wizualizacja i analiza macierzy połączeń



	A	B	C	D	E
A					
B					
C					
D					
E					

Slajd z wykładu Alex Bigelow: Graphs and Trees, University of Utah, 2014

Wizualizacja i analiza macierzy połączeń

Wykorzystywane dla gęstych grafów

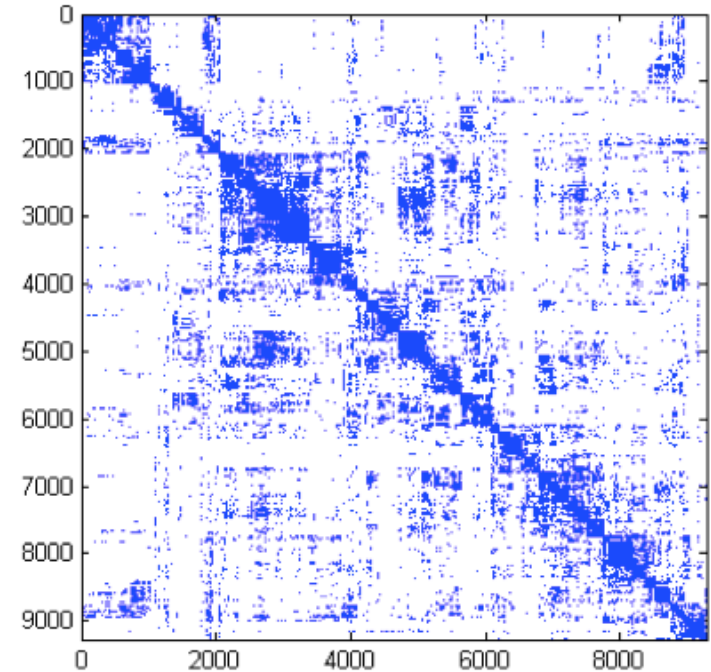
Pozwalają dostrzec

- Regularności
- Pod-grafy

Uporządkowanie wierszy wpływa na
wynik wizualizacji

Wymaga doświadczenia od odbiorcy

Niełatwe śledzenie ścieżek w grafie



Slajd z wykładu Alex Bigelow: Graphs
and Trees, University of Utah, 2014

Trees – wizualizacja drzew

Node-link diagrams

Węzły rozmieszczone równomiernie w przestrzeni 2D, dopóki rozmiar pozwoli

Typowo tzw. drzewo płaskie

Także nietypowe reprezentacje

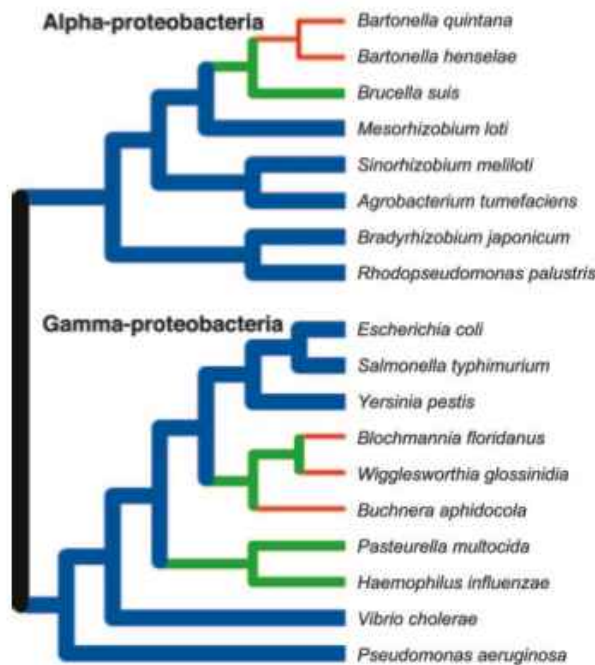
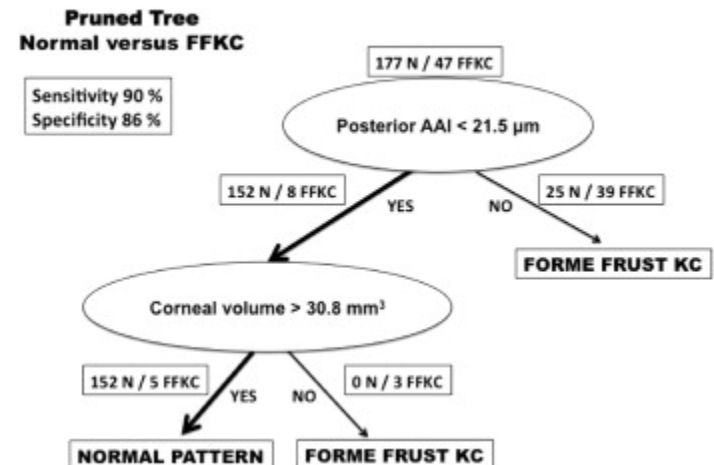
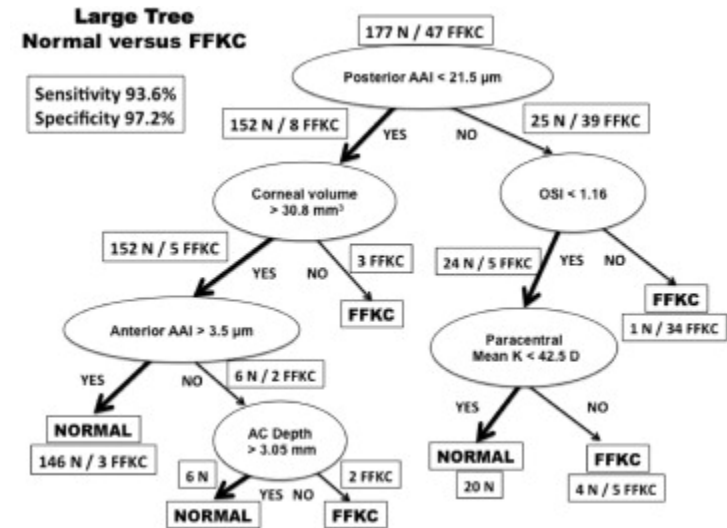
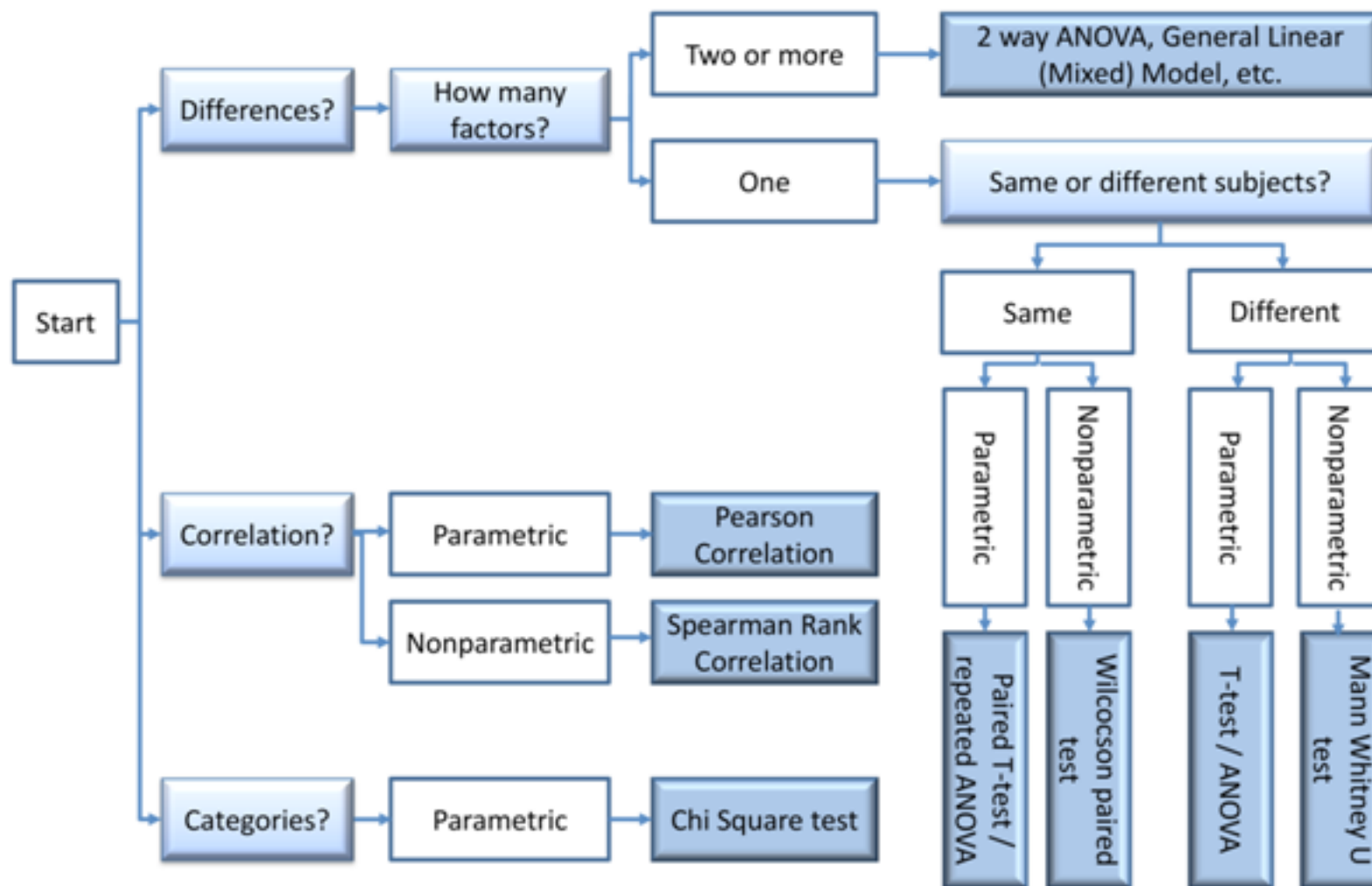


Fig. 1. Relationships of sequenced bacterial genomes showing that in both the α - and γ -proteobacteria, lineages with smaller genome sizes are derived from ancestors that had larger genomes. Branch widths and colors correspond to relative genome sizes as follows: red, <2 Mb; green, 2-4 Mb; blue, >4 Mb. Within the γ -proteobacteria, the insect endosymbionts *Buchnera*, *Wigglesworthia*, and *Blochmannia* form a clade in which all genomes are <700 kb. Differences in the gene repertoires of these symbionts indicate that the extreme reduction in genome size occurred independently after the lineages diverged. Figure adapted from phylogenies presented in refs. 18 and 19.



To jest też diagram typu drzewo



Indentation tree

Umieszczaj elementy w kolejnych wierszach

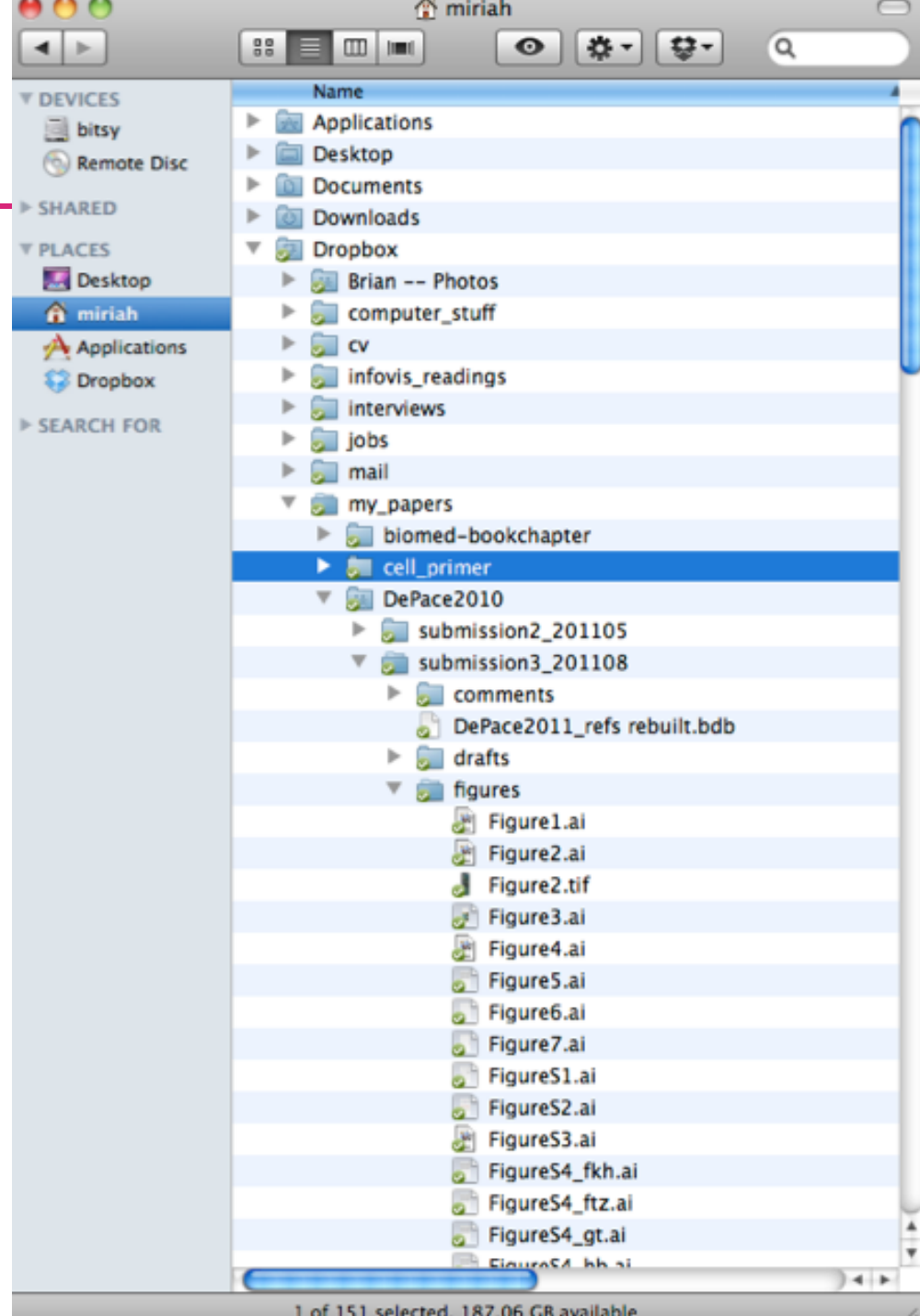
Wcięcie – ma znaczenie:

- Pokazuje zależność rodzic-dziecko

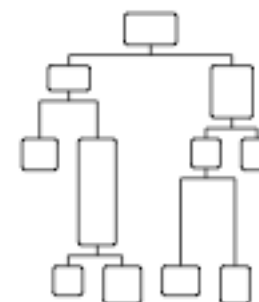
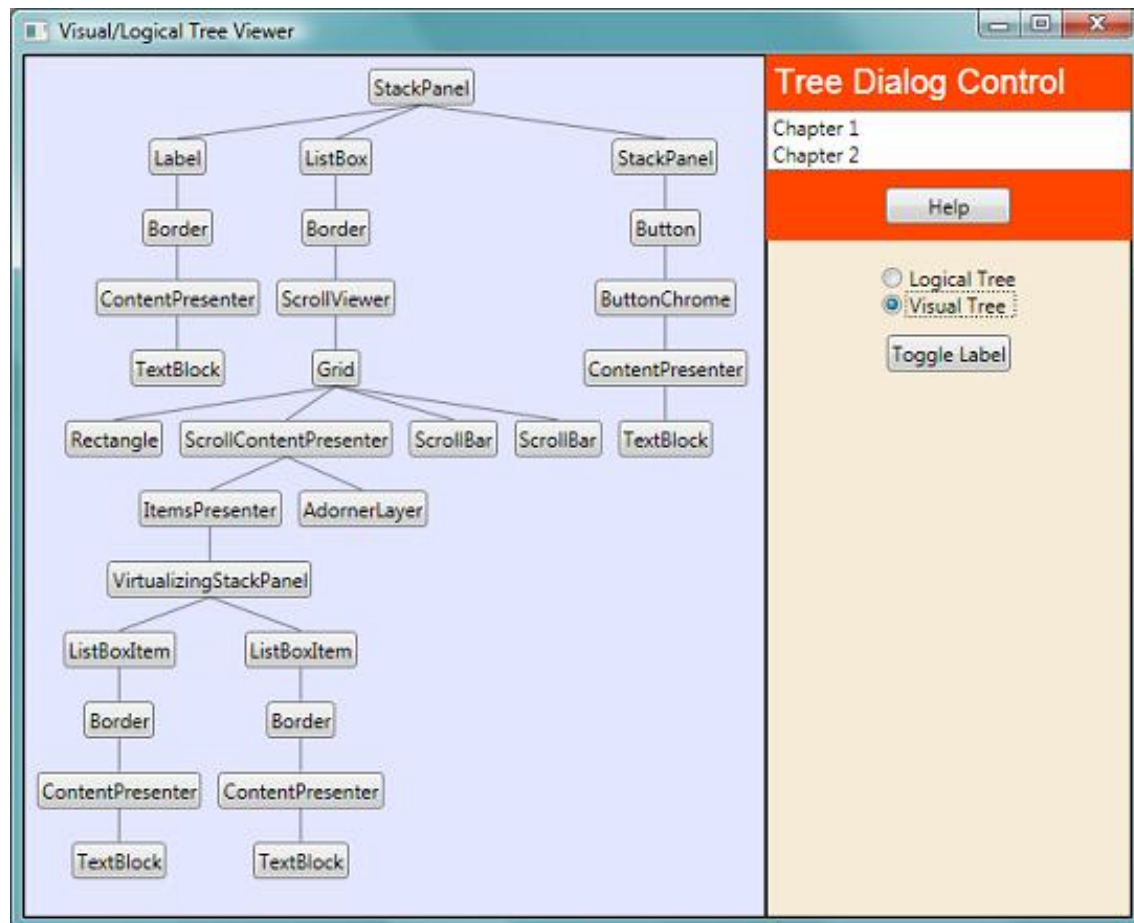
Często stosowany interfejs w informatyce

Może wymagać „scrollowania” ekranu

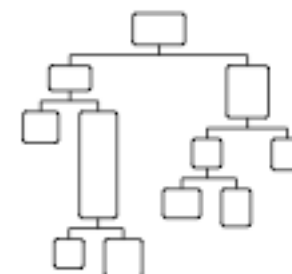
Trudności z wizualizacją dużych drzew



Klasyczne ułożenie - algorytm Reingold and Tilford



(a) Layered.

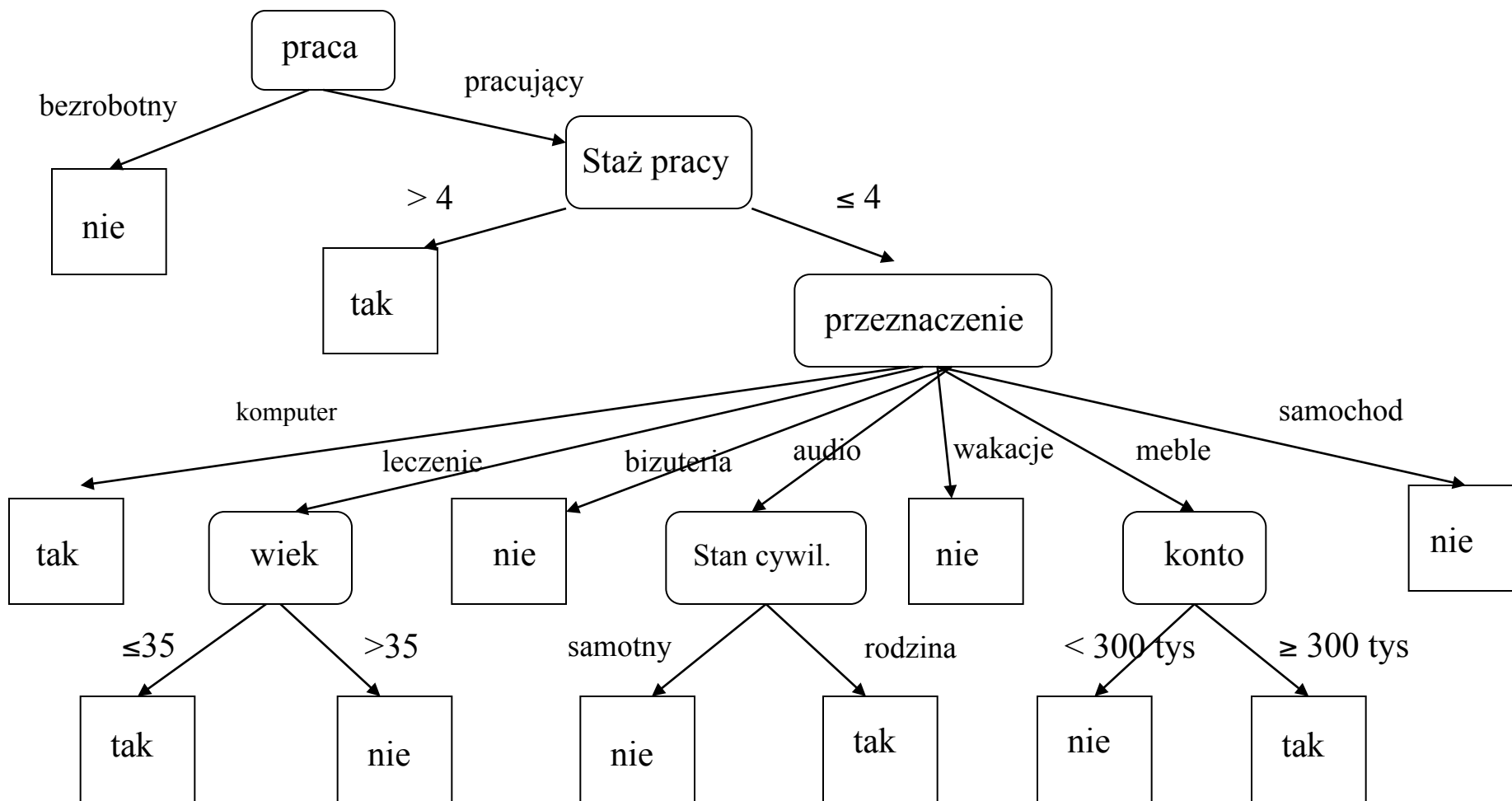


(b) Non-layered.

Przykład eksploracji danych - Japanese credit data

- W przykładzie wykorzystano dane dotyczące 125 osób ubiegających się o kredyty konsumpcyjne w pewnym banku w Japonii (archiwum Univ. Irvine).
- Osoby scharakteryzowane za pomocą 10 cech jakościowych i ilościowych, np. sytuacja zawodowa, przeznaczenie kredytu, płeć, stan cywilny, wiek, zarobki, stan konta, deklarowane raty, staż pracy w zakładzie (lata), ...
- Klienci byli podzielenie na dwie grupy: dobrych (mogą uzyskać kredyt) i ryzykownych.
- Cel analizy: identyfikacja reguł decyzji kredytowych, poszukiwanie profilu klientów którzy nie powinni otrzymać kredytu

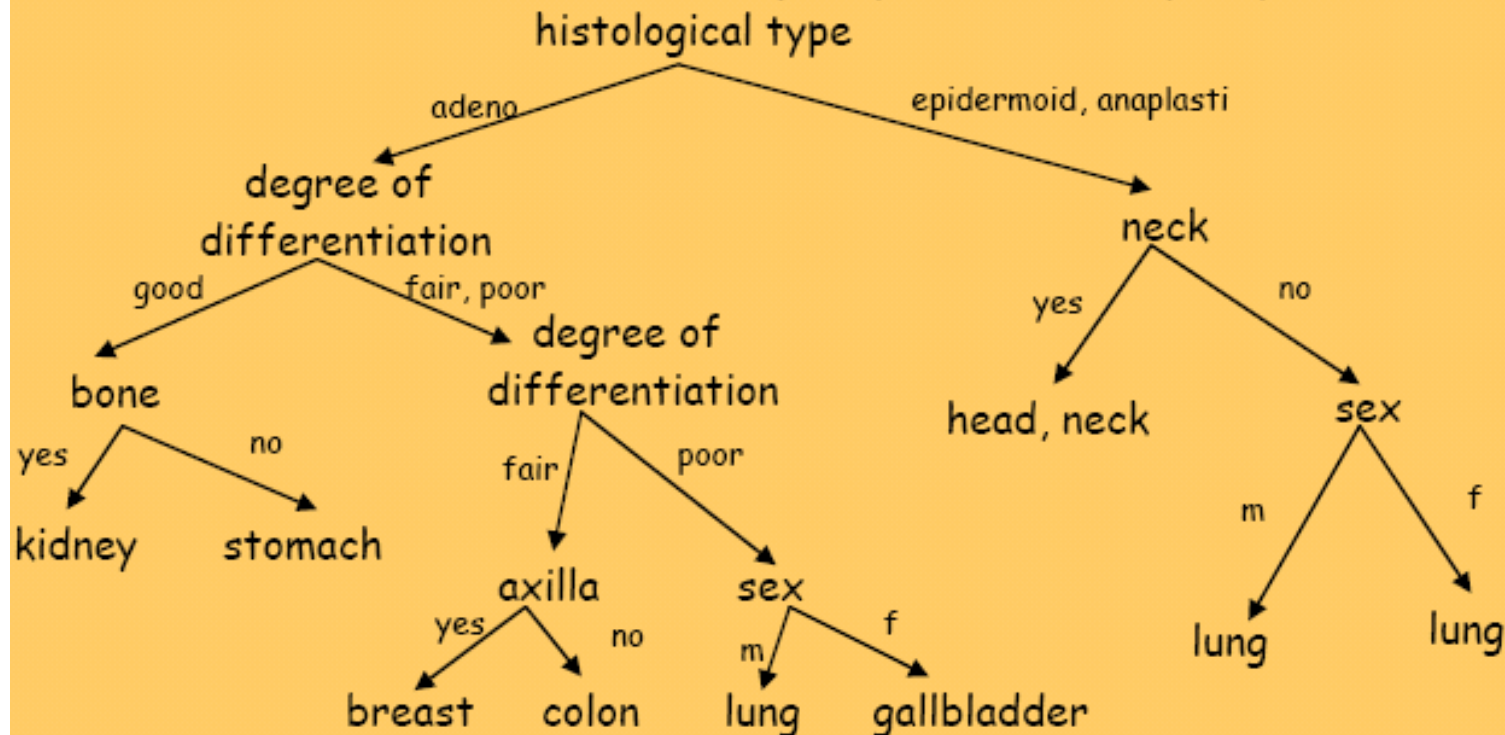
Final tree for credit data



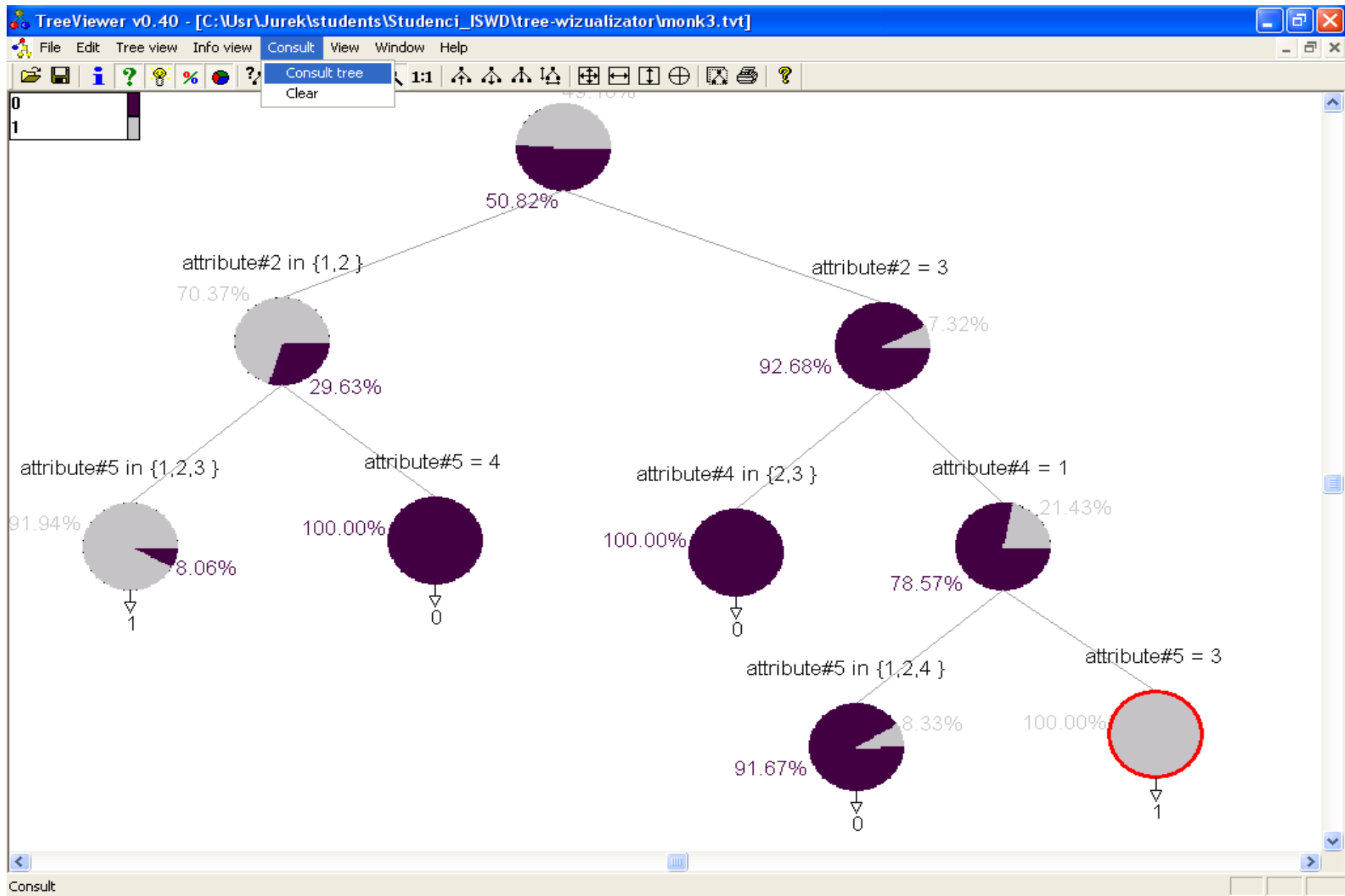
Indukcja drzew z danych medycznych U.Lubljana

Location of primary tumor

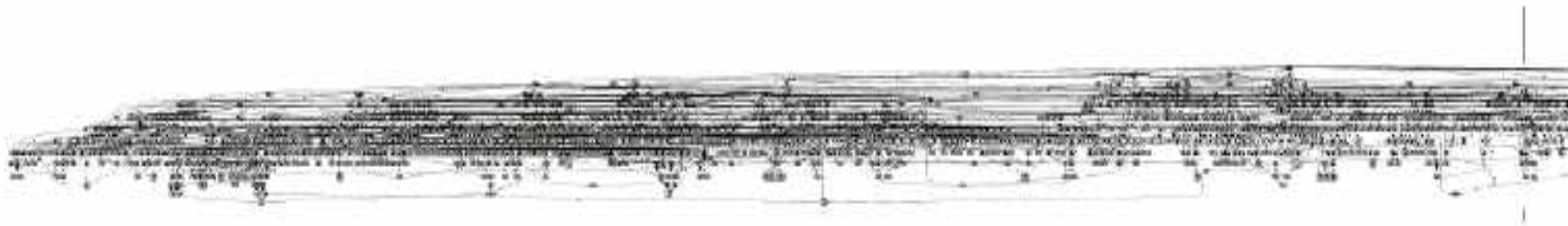
- 339 examples
- 228 for learning, 111 for testing
- induce decision tree accuracy: unpruned: 41%, postpruned: 45%



Tools for decision tree visualisation



Redukcja, upraszczanie drzewa (tree pruning)

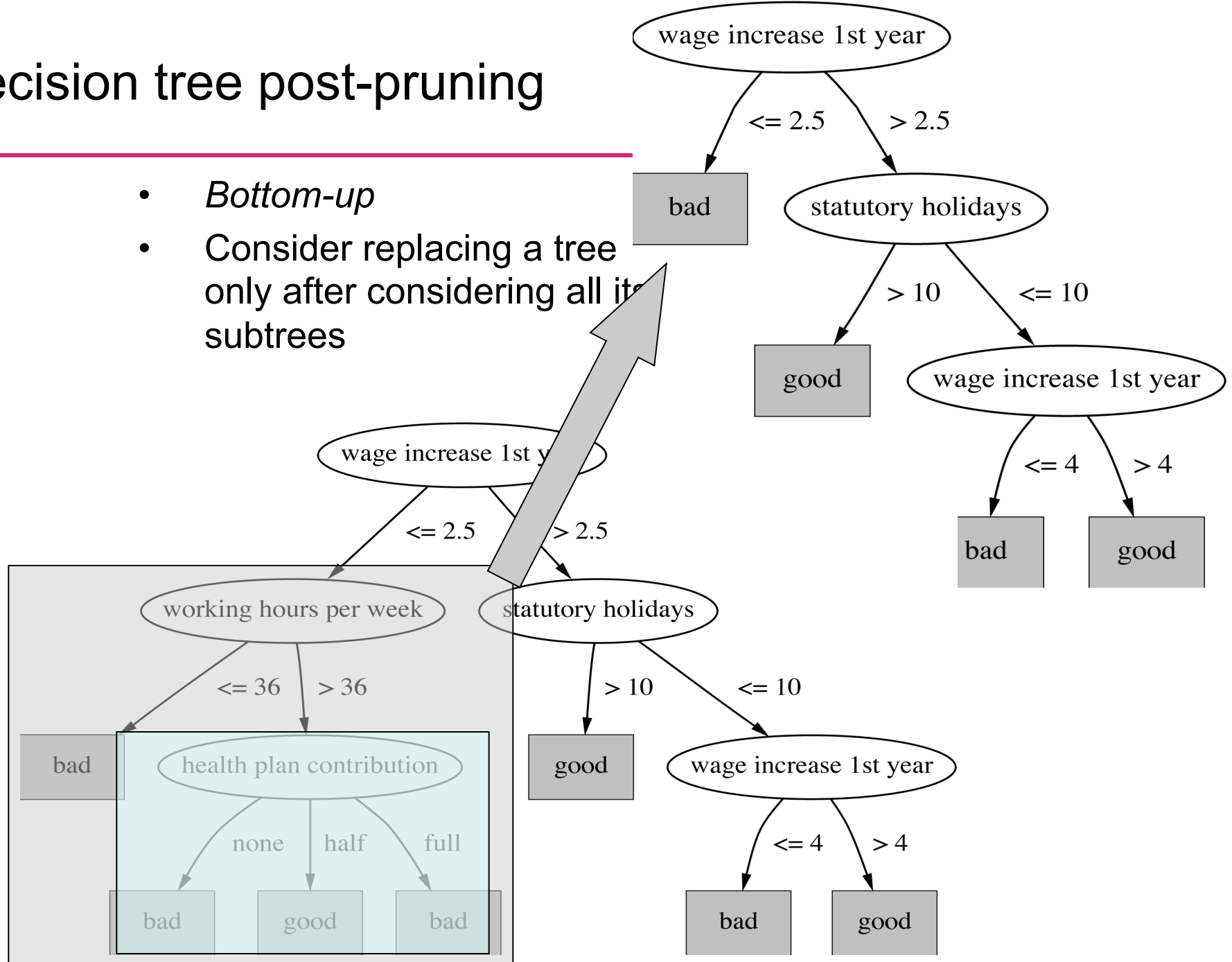


Wiele drzew (zwłaszcza automatycznie odkrywanych z danych) ma zbyt duże rozmiary dla analizy przez człowieka:

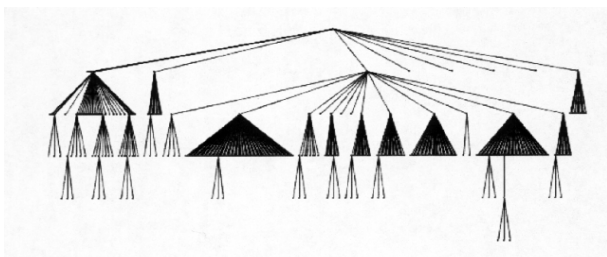
- Dotyczy zarówno zadań indukcji drzew decyzyjnych jak i hierarchicznych algorytmów analizy skupień
- Wyzwanie dla wielkich i złożonych danych
- Pomimo symbolicznej reprezentacji – trudność dostrzeżenia szczegółów
- Klasyczne algorytmy indukcji drzew – najważniejsze atrybutu blisko korzenia drzewa
- Rozwinięte techniki redukcji drzew (pre-pruning vs. post-pruning)

Decision tree post-pruning

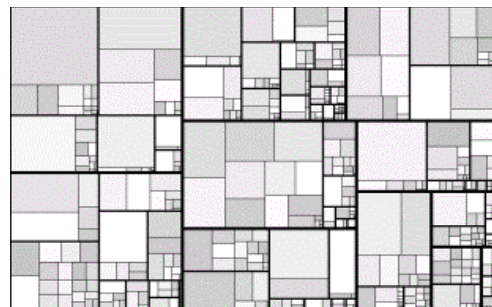
- *Bottom-up*
- Consider replacing a tree only after considering all its subtrees



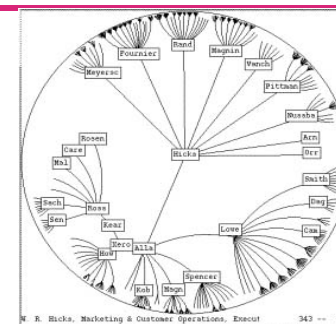
Trees – różne reprezentacje



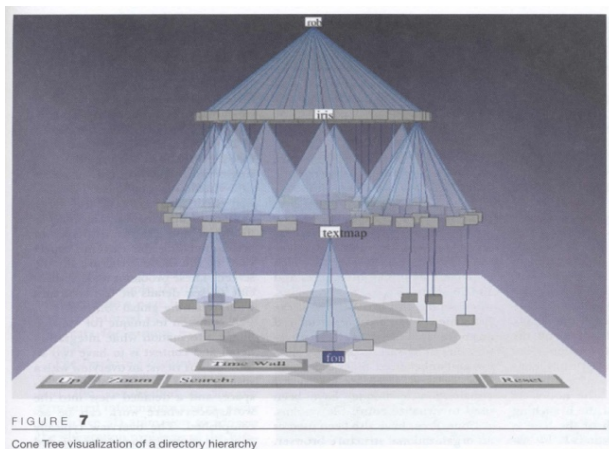
Traditional



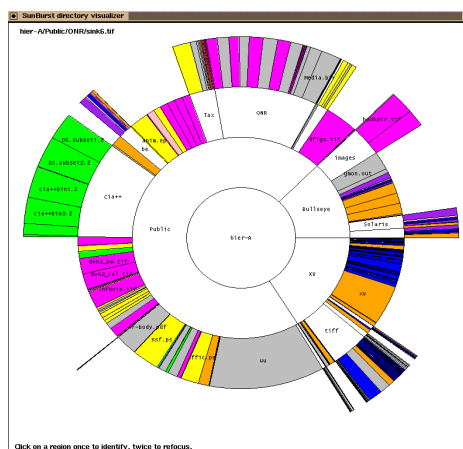
Treemap



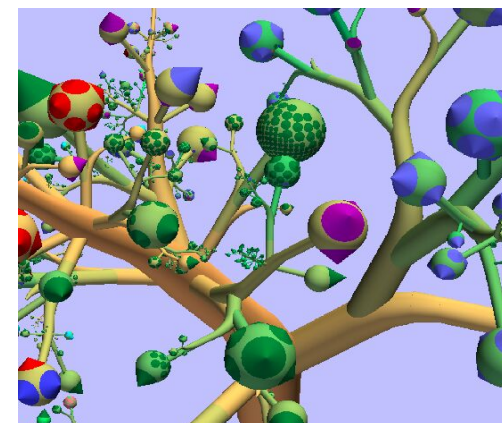
Hyperbolic Tree



ConeTree



SunTree



Botanical

Zmiana układu współrzędnych – polar coordinates

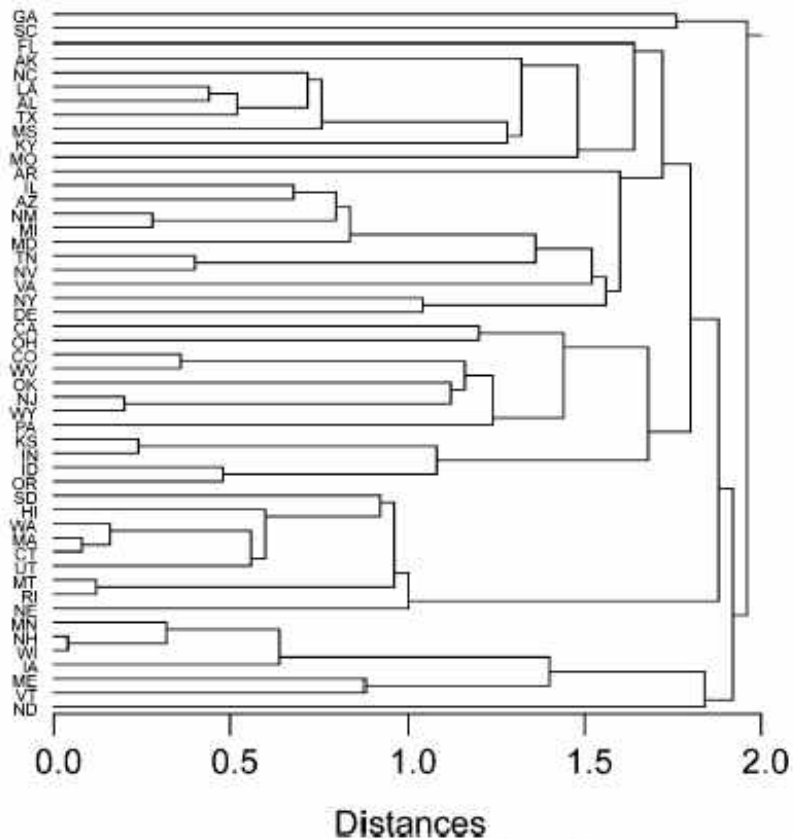


Figure 5.4. Hierarchical cluster tree of US murder rates

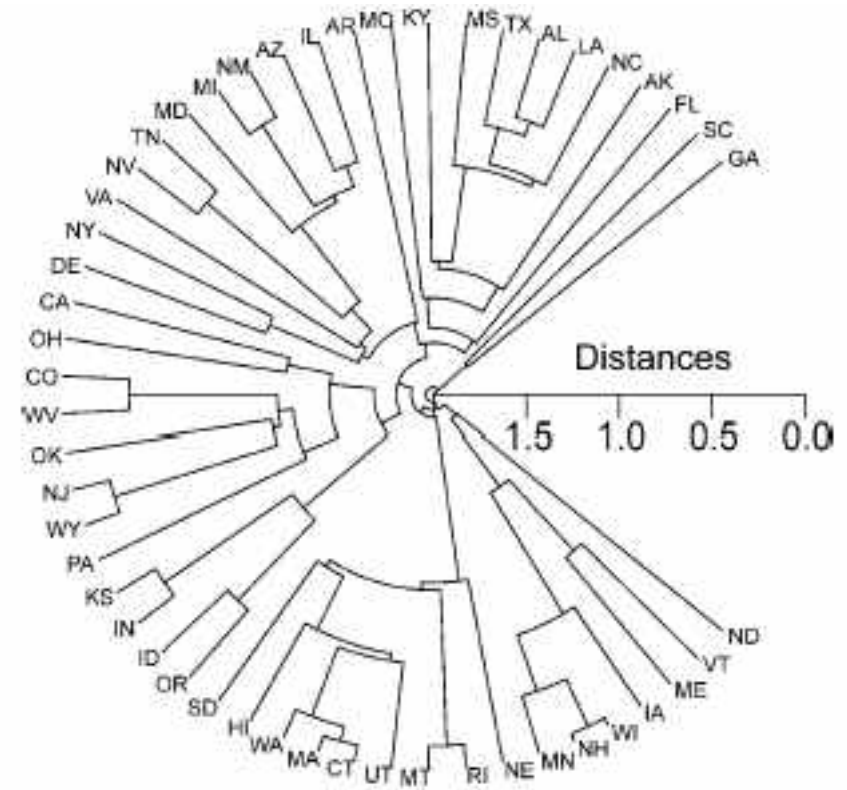


Figure 5.5. Polar hierarchical cluster tree of US murder rates

Inne wizualizacje

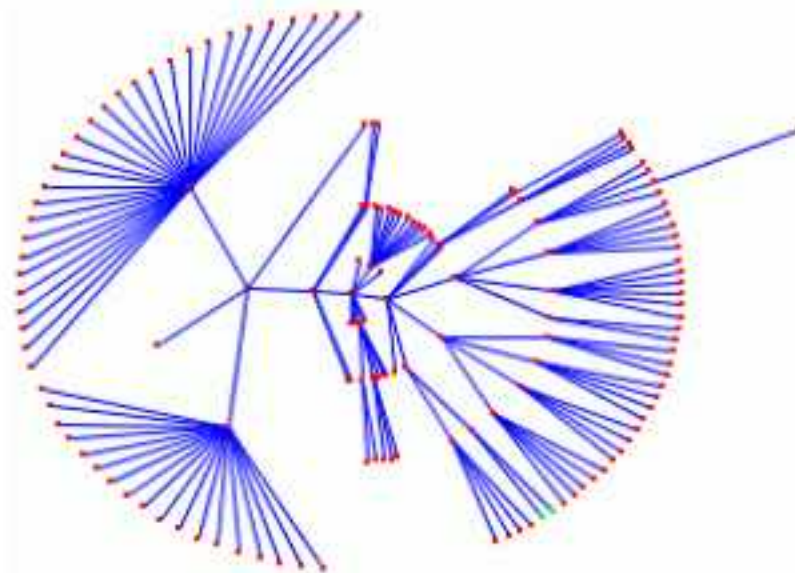


Fig 4. Radial View

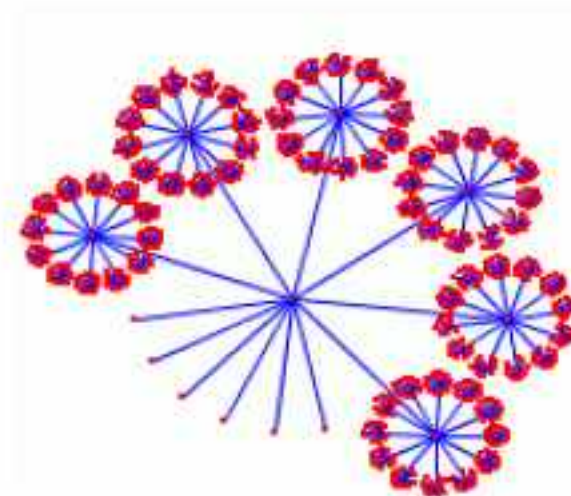
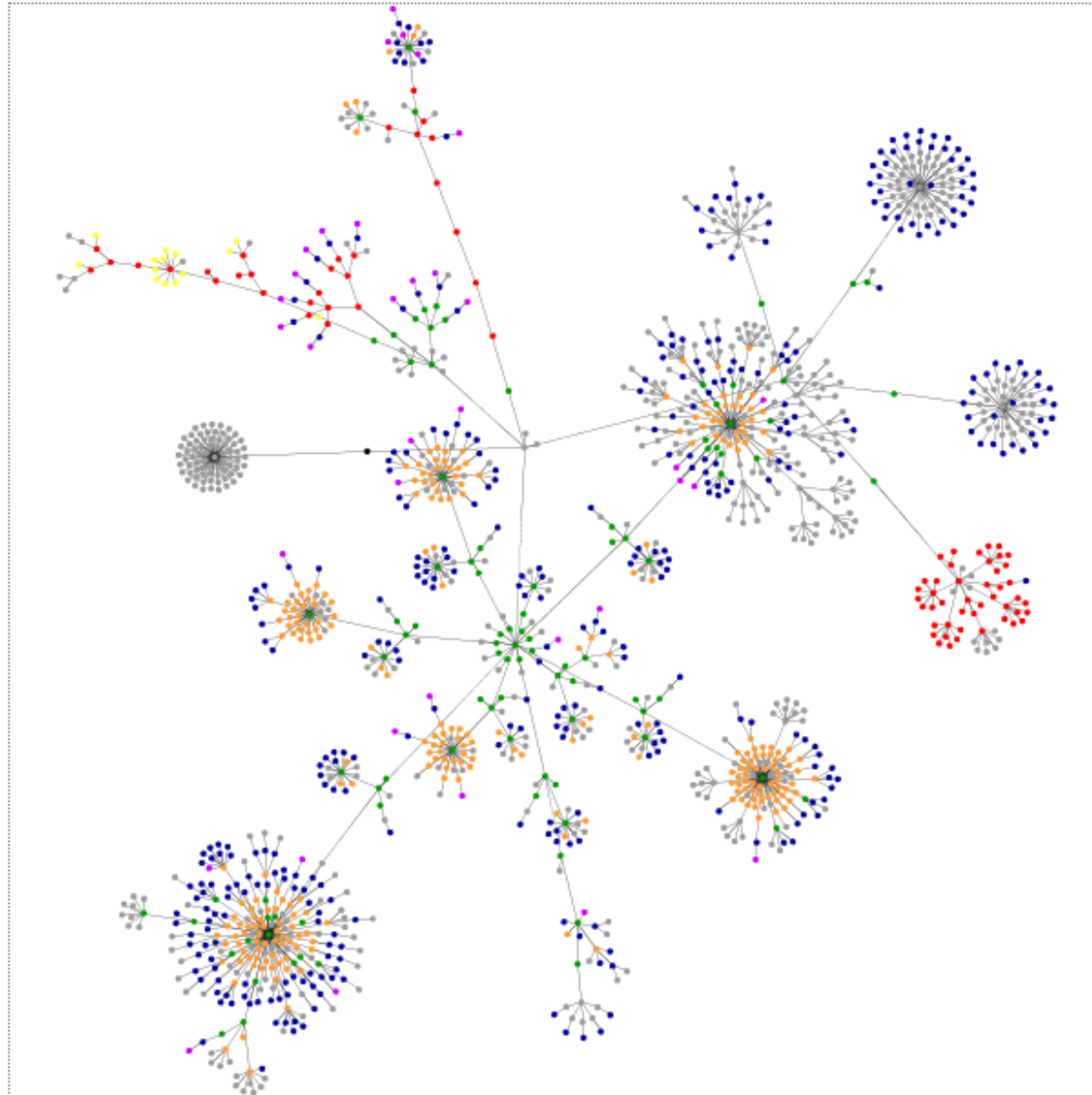
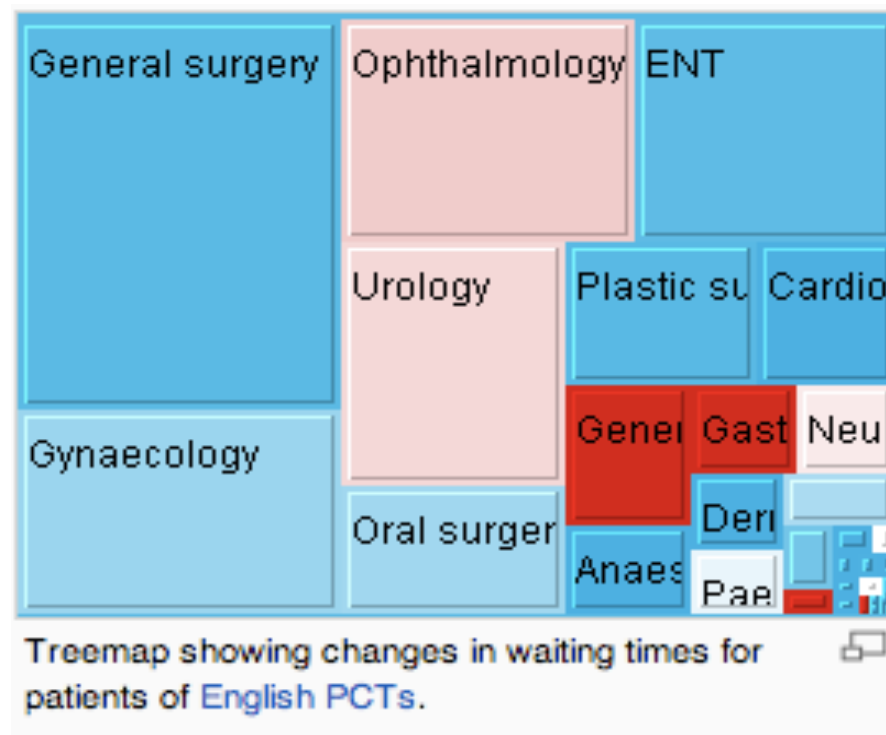


Fig 5. Balloon view

Wizualizacja sieci WWW



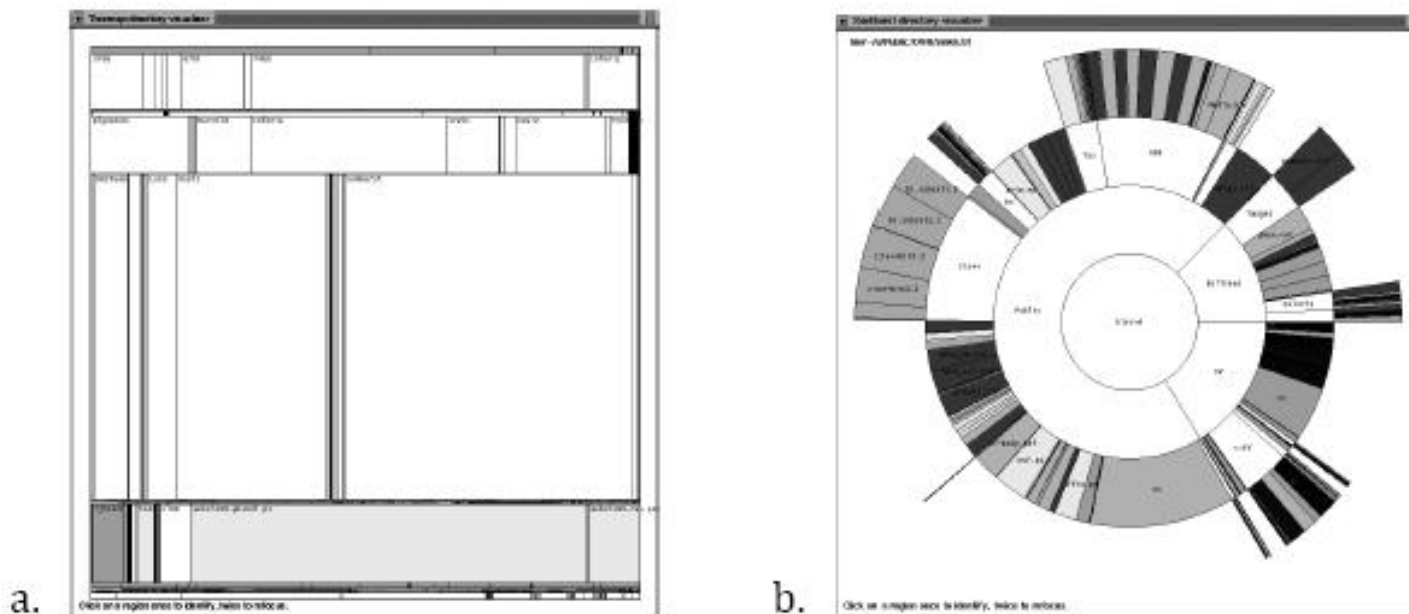
Hierarchiczne wizualizacje – Treemaps (Johnson and Shneiderman 1991)



- Treemaps display hierarchical data using rectangles. Each branch of the tree is assigned a rectangle. Then each sub-branch gets assigned to a rectangle and this continues recursively until a leaf node is found.
- A leaf node's rectangle has an area proportional to a specified dimension (cardinality) on the data. (Illustration -- proportional to a waiting time)
- Depending on choice the rectangle representing the leaf node is colored, sized or both according to chosen attributes.

Prostokąty vs. hierarchia pierścieniowa

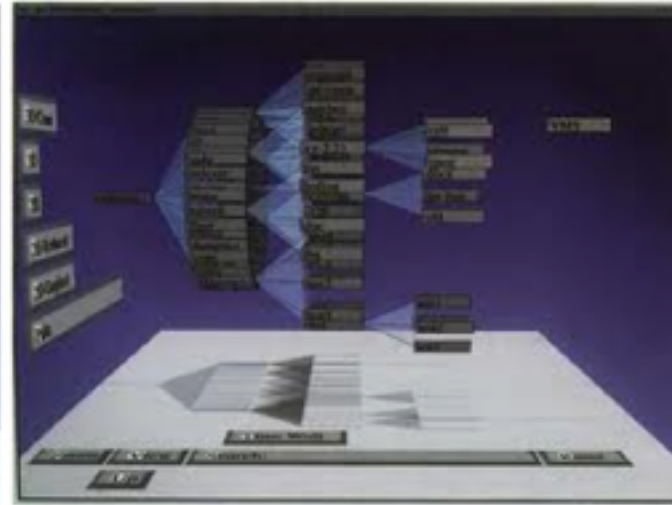
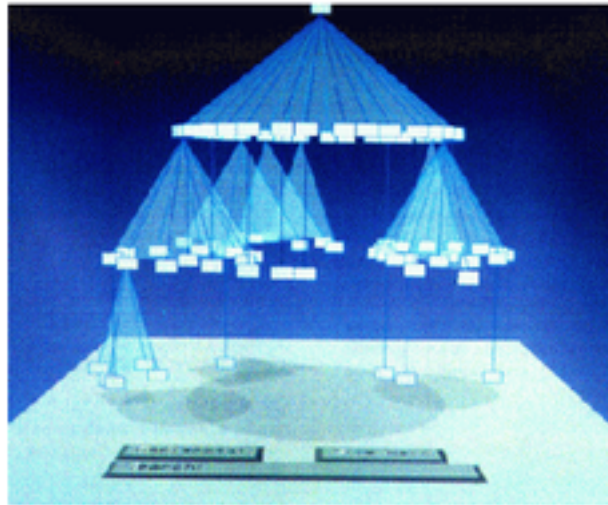
Cel – lepsza wizualizacja zależności hierarchicznej



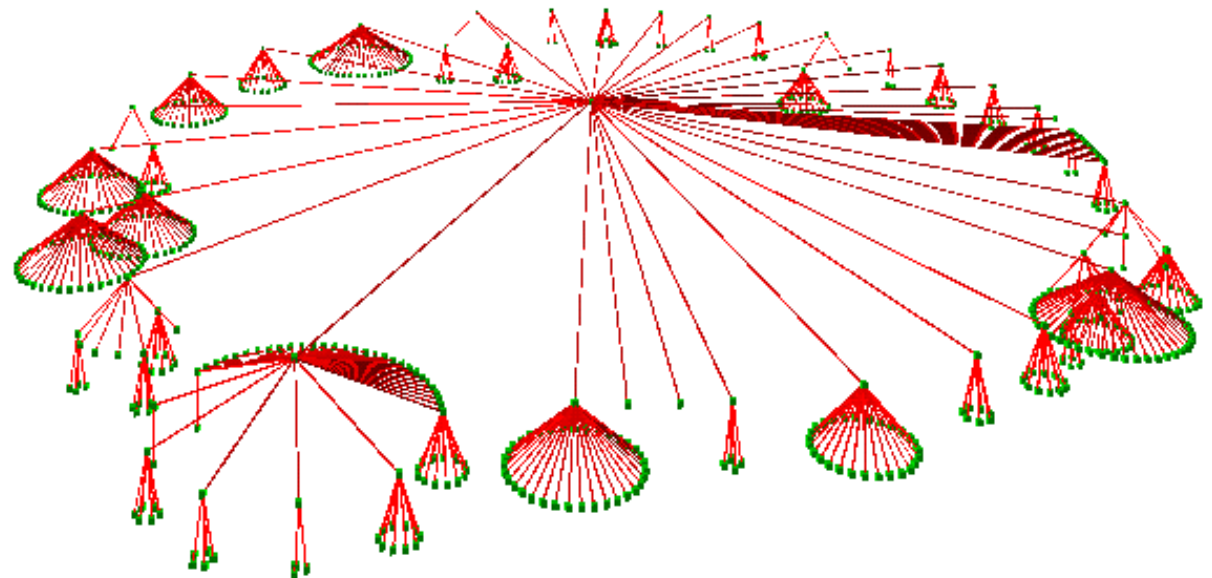
Rysunek 1. Strategia prostokątna (a) i pierścieniowa (b) wizualizacji zasobów katalogowych

Źródło: J. Stasko, R. Catrambone, M. Guzdial, K. McDonald, *An evaluation of space-filling information visualizations for depicting hierarchical structures*, „International Journal of Human-Computer Studies” 2000, vol. 53, iss. 5, s. 667–668.

Cone trees



Cone Tree - Information Visualizer - Xerox Parc from *Cone Trees: Animated 3D Visualizations of Hierarchical Information* George G. Robertson, Jock D. Mackinlay, & Stuart K. Card 1991



Hyperbolic view of the tree – 3D

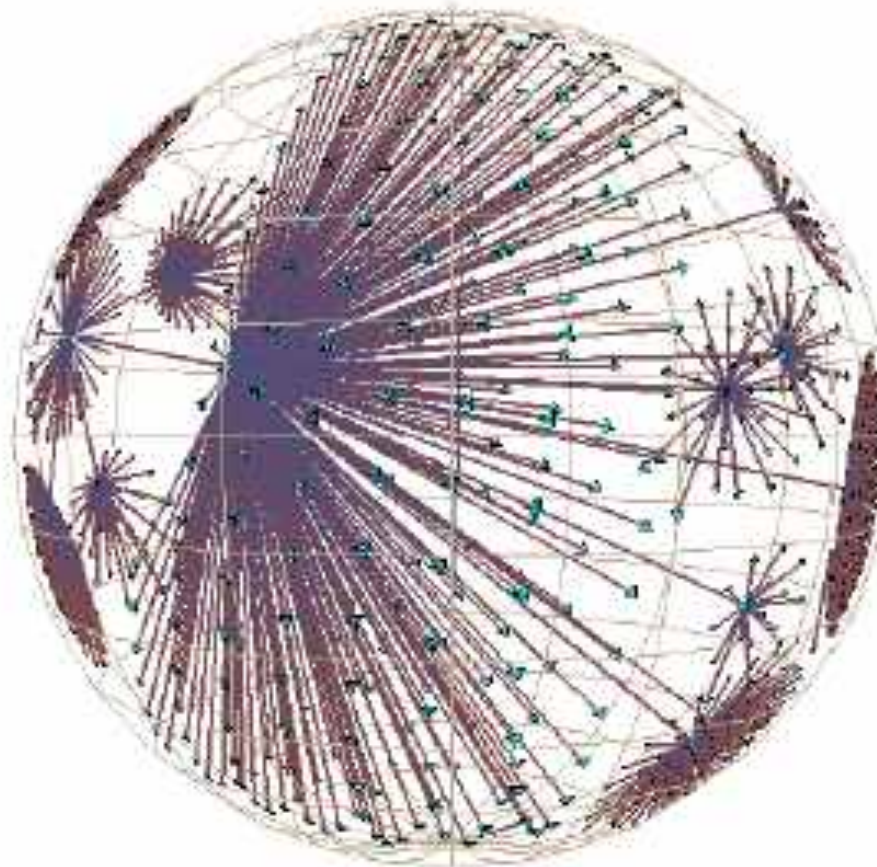


Fig 11. Hyperbolic view of a tree in 3D. (Courtesy of T. Munzner, Stanford University, USA.)

Narzędzia wspomagające rysowanie grafów

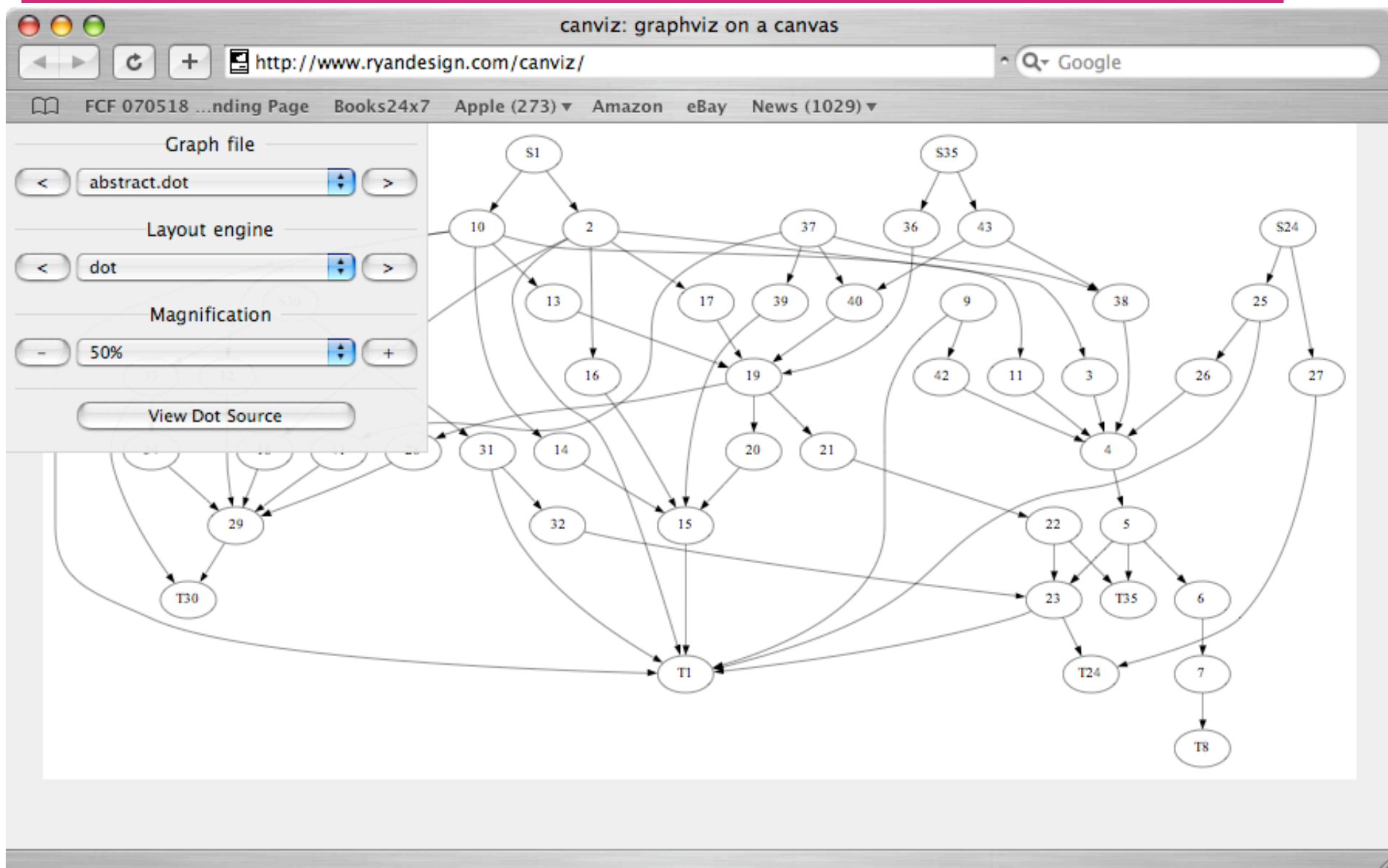
Popularne oprogramowanie:

- Prefuse (Information visualization software)
- Gephi (Graph visualization software)
- GUESS (Interactive interface to modify graph properties)
- Cytoscape
- GraphViz
- NetDraw

Specjalistyczne biblioteki dla Big Data

np. Pegasus

Graphviz software



Literatura

- Chen, C.H., Härdle, W. and Unwin, A. (eds). *Handbook of Data Visualization*, Springer, Heidelberg (2008).
- Unwin, Theus, Hofmann: *Graphics of Large Datasets. Visualizing a Million*, Springer (2006)
- Graph Visualisation and Navigation in Information Visualisation: a Survey. Ivan Herman, Guy Melancon, M. Scott Marshall. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24-44, 2000.

Także materiały wykładowe:

- Chris North: cs5984 Information Visualization (Web site: www.cs.vt.edu/~north/infoviz/)
- ...

