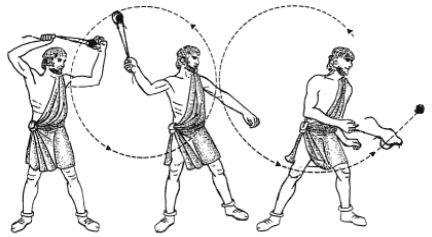
AllegroGraph for Social Network Analysis

AllegroGraph as an event database with social network analysis and geospatial and temporal reasoning

Today: main focus
Social Network Analysis (SNA)

- What is new in AllegroGraph 3.0 (< 5 minutes)
- SNA is part of 'event data base' story
- Questions that SNA addresses
- Some technical details about our SNA
- Demo 1: A database of friendship and love
- Demo 2: DBpedia
- Visit us at SemTech, May 19-22, San Jose
- JavaOne: May 8th, Semantic Panel 1:30PM

AllegroGraph as graph store



- Scalable and persistent quadstore
  - 10 Billion quads in a day on affordable hardware
- Relational database efficiency for range queries
  - We support most xml schema types (dates, times, longitudes, latitudes, durations, telephone numbers, etc)
- Compliant on standards
  - RDF, RDFS, OWL, SPARQL, Named Graphs, ISO Prolog
- Standalone and Client/Server
  - Direct Socket Interface: Java (Sesame/Jena), Lisp,
  - REST interface, Ruby, C
- Reasoning: OWL subsets and full Description Logics
- GUI & Ontology Management: Top Braid Composer & Racer Porter
- Full Text indexing: Google the content of your triple store.
- Named Graphs fully supported: Slot used for weights, trust factors, provenance, distance, etc.

New in AllegroGraph 3.0

- **Federation:**
  - **Create an abstract store as collection of other triple stores.**
  - **Transparently use Prolog and SPARQL and Reasoning**
- **Spatial database efficiency for geospatial primitives**
  - **Find elements in bounding boxes as fast as in spatial databases, basic polygon handling**
- **Temporal reasoning**
  - **Reasoning about times and intervals (Allen Logic)**
- **Social Network Analytics library**
  - **Find actor degrees, cliques, actor centrality, group centrality and group cohesiveness.**
- **GRUFF: our new navigation tool for large triple stores**
- **Stay tuned for Webinars on**
  - **Oracle connection, Python, Ruby and C# interfaces**

Download Learning Center Search Community Resources

AllegroGraph

RacerPro

TopBraid Composer

Services

Support

Learning Center

Documentation

Updates

- Info

FAQ

About Franz Inc.

XML

powered by AllegroServe

AllegroGraph Java Edition Tutorial Examples

This Learning Center is designed to facilitate understanding of RDF database technologies and best practices for AllegroGraph. It contains examples for working with RDF triples, Triple Stores and Server Management, Querying with SPARQL, and Reasoning with RDFS++ and Prolog. The software examples are freely downloadable as an Eclipse project java archive and work with the Java version of AllegroGraph, including the Free Edition.

Please send any comments or suggestions to info@franz.com.

Preamble and Installation

- Choosing an Edition
- Downloading the software
- Installing the software
- Updating an installation

Running the Examples

- Starting a server manually
- Building in Eclipse
- Running in Eclipse
- Configuring Paths
- Building from a Command Line
- Running from a Command Line
- Stopping a server manually

Server Management

- Connecting to a server
- Getting server parameters

Indexing a Triple Store

- Overview of Indexing
- Indexing all triples
- Indexing new triples
- Automatic Indexing
- Index Flavors
- Free Text Indexing

Basic Retrieval

- Matching Triples
- Working with Cursors
- Range Queries
- Free Text Search

SPARQL

- Overview of SPARQL
- Basic Graph Patterns
- Matching RDF Literals

Social Network Analysis

- Generators
- Degrees and Neighbors
- Search Paths
- Metrics and Centrality
- Cliques

Geospatial Primitives

- Cartesian Data
- Spherical Data
- Cartesian Bounding Boxes
- Spherical Bounding Boxes
- Haversine Distance
- Polygon Regions
- Predicate Mappings
- Loading and Serializing

Temporal Primitives

Events and Activity recognition

- Our customers use AllegroGraph as an event database with social network analysis and geospatial and temporal reasoning

Find all meetings that happened in May within 5 miles of Berkeley that was attended by the most important person in Jans' friends and friends of friends.

The common elements of an event

- A type
  - Meetings, communications event, financial transactions, visit, attack/truce, an insurance claim, a purchase order
  - Reasoning over types of events requires RDFS++ reasoning
- A list of actors
  - Reasoning over relationships between actors requires Social Network Analysis
- A place
  - Reasoning over where something happened or how far away something happened requires GeoSpatial Reasoning
- A Start-time and possible an end-time
  - Reasoning over when or in what order something happened requires Temporal Reasoning
- Anything else that describes the event
  - Goods that changed hands

Events at the core of many Business Processes

- Health care
  - A hospital visit, a visit to a drugstore, a medical procedure
- Communications Industry
  - A telephone call (and they store your location now too)
  - An Email or SMS
- Financial industry
  - Every financial transaction is an event
- The insurance industry
  - Track behavior of customers & find fraud
  - Predict calamities and pay offs.
- The enterprise
  - Combine your ERS system, your email archive and your HR data
- The Government
  - HS is interested in every type of imaginable event
  - Entering/leaving the country, going into/out of hotels, every telephone call and email, every payment done, every trip made, etc..

Integrated in query language.

Find all meetings that happened in May within 5 miles of Berkeley that was attended by the most important person in Jans' friends and friends of friends.

```
(select (?x)
   (ego-group !person:jans knows ?group 2)
   (actor-centrality-members ?group knows ?x ?num)
   (q ?event !fr:actor ?x)
   (qs ?event !rdf:type !fr:Meeting)
   (interval-during ?event "2008-05-01" "2008-05-07")
   (geo-box-around !geoname:Berkeley ?event 5 miles)
   !)
```

Integrated in select language.

Find a meetings that happened in May within 5 miles of Berkeley that was attended by the most important person in Jans' friends and friends of friends.

```
(select (?x)
   (ego-group !person:jans knows ?group 2)
   (actor-centrality-members ?group knows ?x ?num)
   (q ?event !fr:actor ?x)
   (qs ?event !rdf:type !fr:Meeting)
   (interval-during ?event "2008-05-01" "2008-05-07")
   (geo-box-around !geoname:Berkeley ?event 5 miles)
   !)
```

Integrated in select language.

Find a meetings that happened in May within 5 miles of Berkeley that was attended by the most important person in Jans' friends and friends of friends.

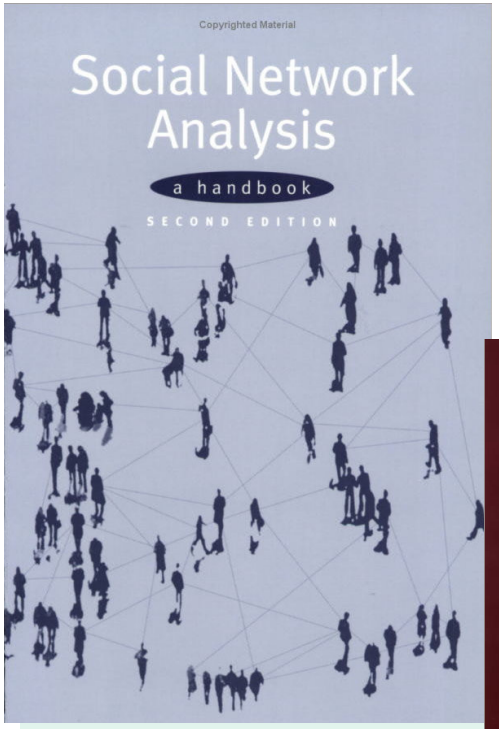
```
(select (?x)
   (ego-group !person:jans knows ?group 2)
   (actor-centrality-members ?group knows ?x ?num)
   (q ?event !fr:actor ?x)
   (qs ?event !rdf:type !fr:Meeting)
   (interval-during ?event "2008-05-01" "2008-05-07")
   (geo-box-around !geoname:Berkeley ?event 5 miles)
   !)
```

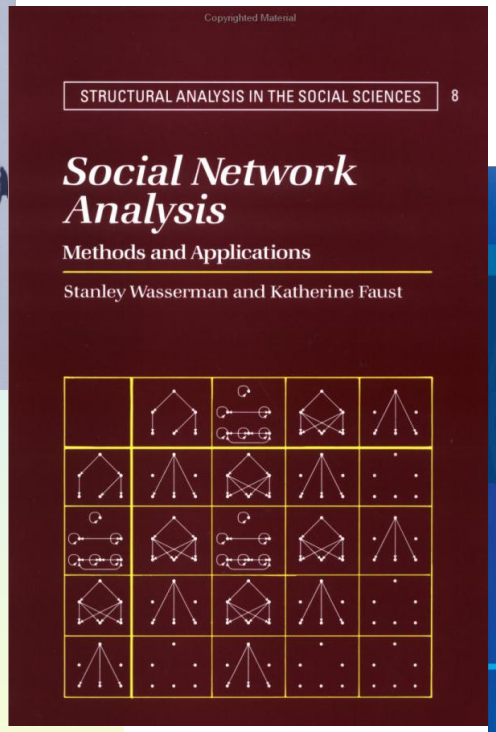
SNA
SNA
DB Lookup
RDFS Reasoning
Temporal
Spatial

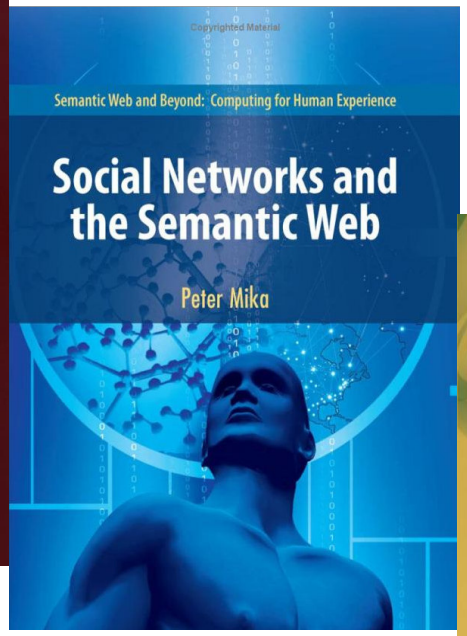
Social Network Analysis (SNA)

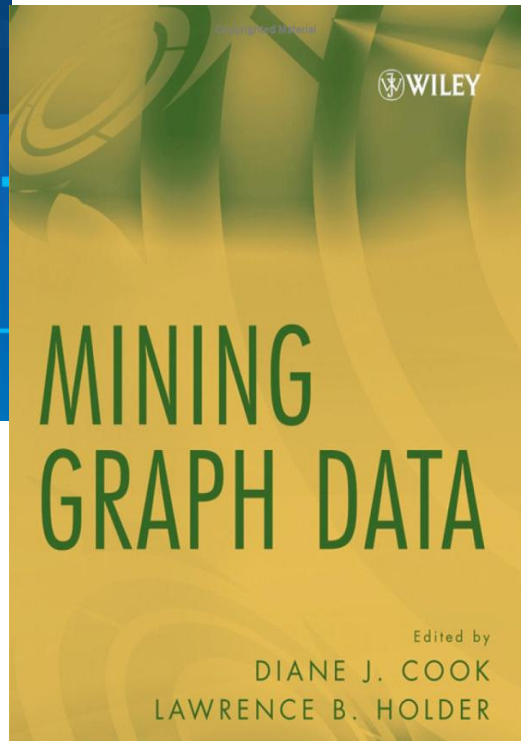
- A good introduction is to be found at
  www.analytictech.com/essex/schedule.htm

1. Introduction to Social Network Analysis	
Topics:	• This course • Overview of field • What is unique about network research
Reading:	• Wasserman and Faust, ch. 1 • http://www.analytictech.com/networks/whatis.htm • http://www.analytictech.com/networks/basic%20concepts%202002.pdf • http://www.analytictech.com/networks/history.htm • http://www.analytictech.com/networks/review_of_empirical_literature.htm • Borgatti & Foster. Borgatti, S.P. and Foster, P.B. The network paradigm in organizational research: A review and typology. [pdf]
Lab:	• Getting acquainted with Ucinet and Netdraw
Optional:	• Freeman, L.C. 1996. Some Antecedents of Social Network Analysis" *Connections* 19: 39-42 [HTML version] [PDF version] • Mayhew, B. 1980. Structuralism vs. individualism prt 1: Shadow boxing in the dark. *Social Forces* :335-375. [IN READING PACKET]
Presentation:	• Introduction.pdf

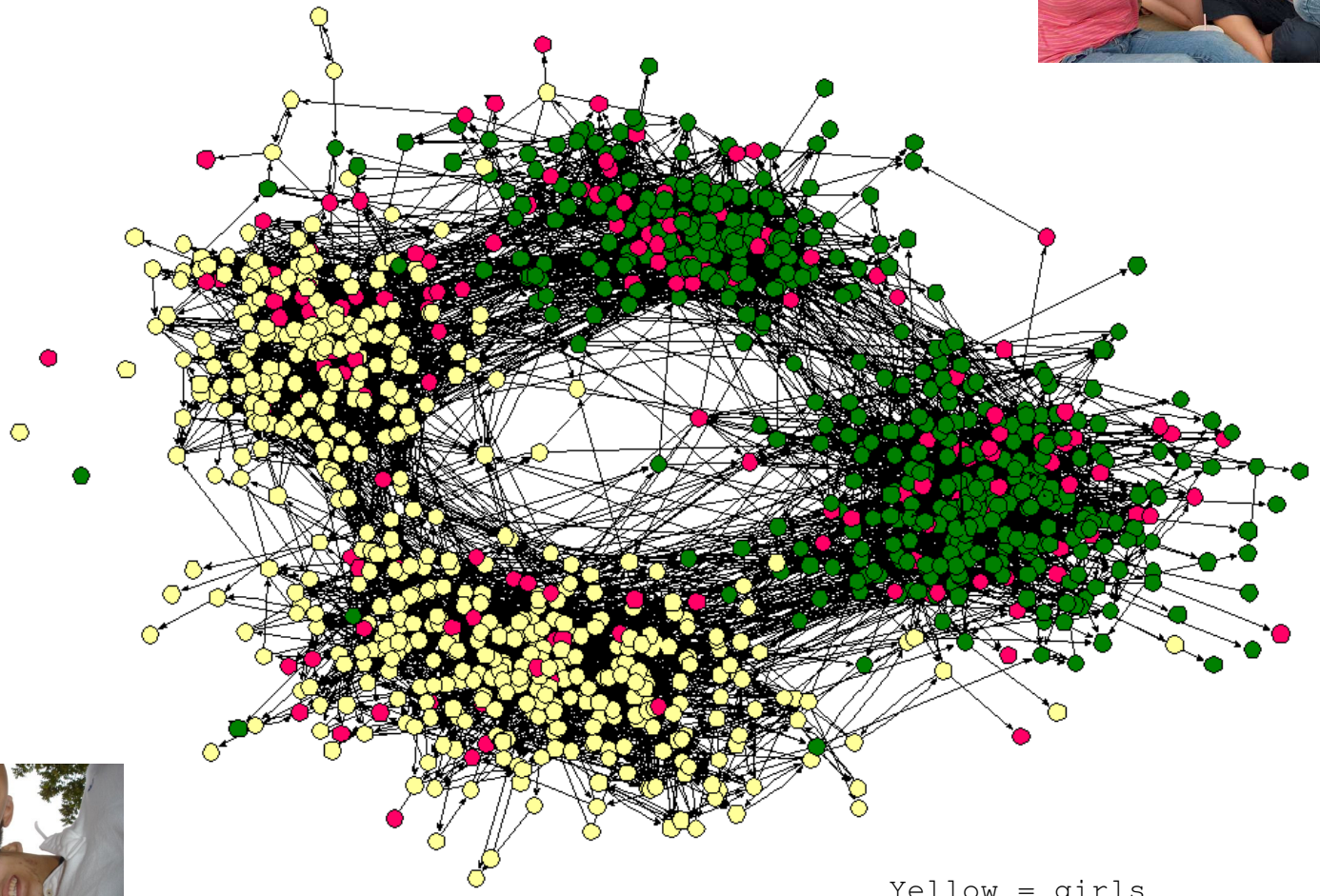

Copyrighted Material
Social Network Analysis
a handbook
SECOND EDITION


Copyrighted Material
STRUCTURAL ANALYSIS IN THE SOCIAL SCIENCES 8
Social Network Analysis
Methods and Applications
Stanley Wasserman and Katherine Faust


Copyrighted Material
Semantic Web and Beyond: Computing for Human Experience
Social Networks and the Semantic Web
Peter Mika


WILEY
MINING GRAPH DATA
Edited by
DIANE J. COOK
LAWRENCE B. HOLDER

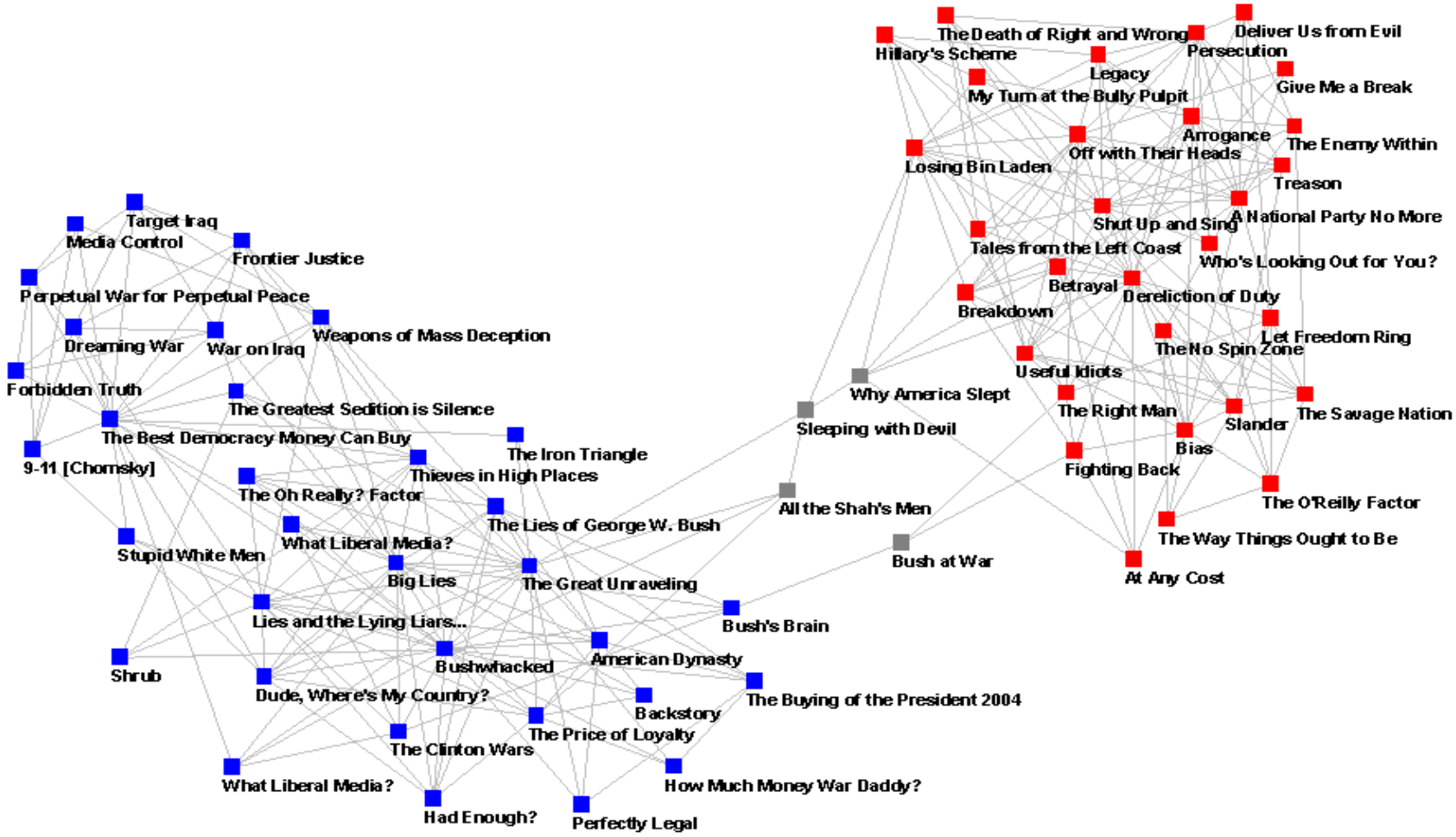
High school friends



Yellow = girls
Green = boys
Red = sexually active

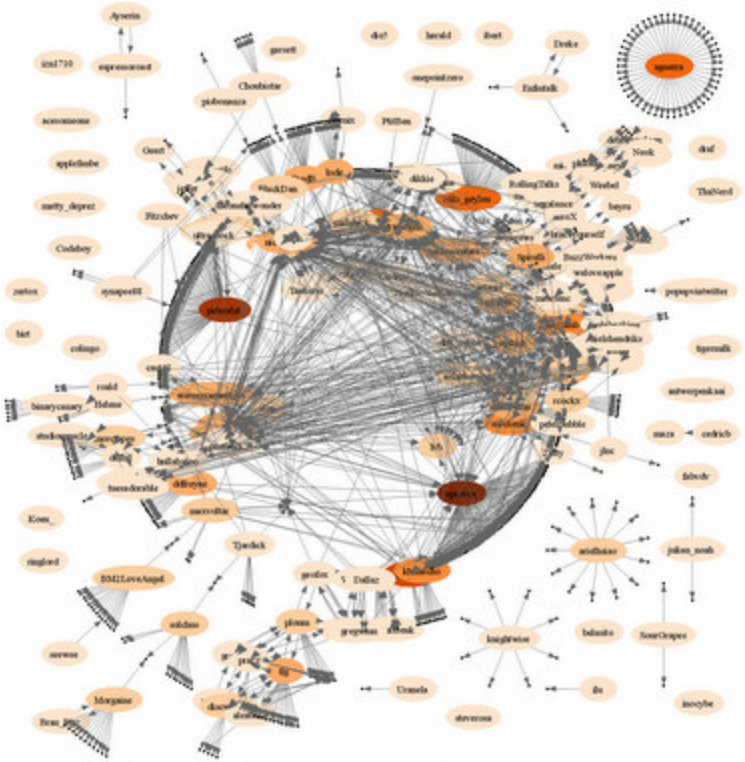
FRANZ INC.
Web 3.0's Database

Political books



Questions in SNA (1)
how far is Actor1 from Actor2?

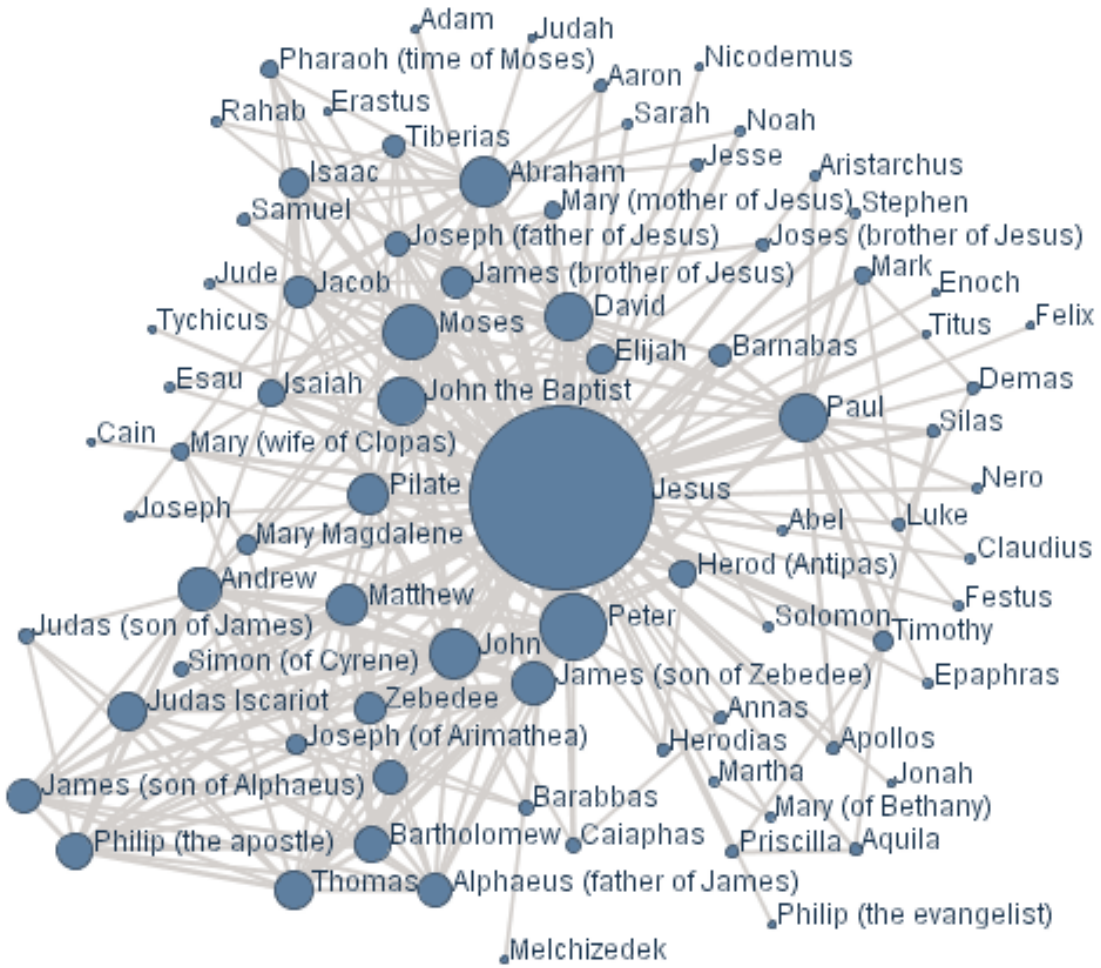
- **Degrees of separation**
  - **How far is P1 from P2**
- **Connection strength**
  - **How many shortest paths from P1 to P2 through a series of predicates and rules**



Twitter Friends van Belgische Twitteraars

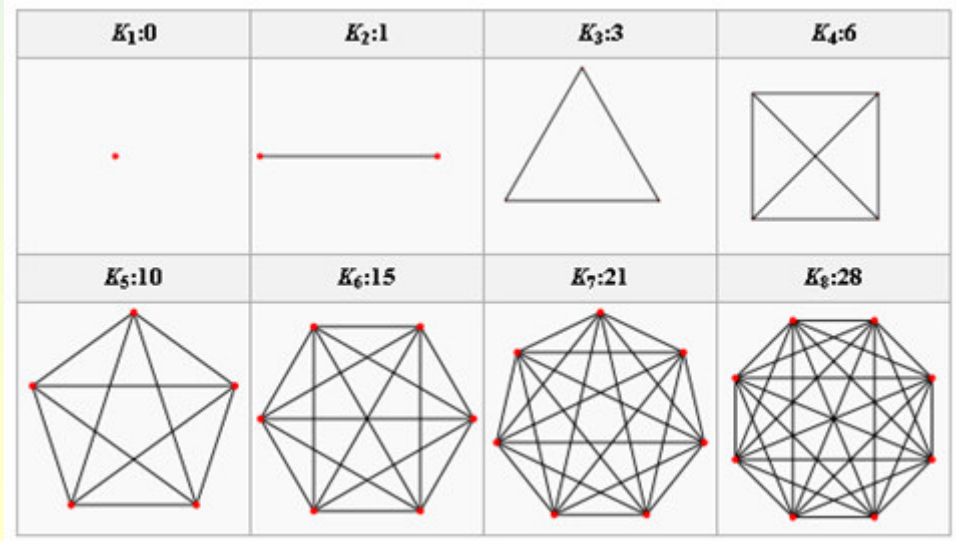
Questions in SNA (2)
What are the groups an actor is in?

- **Find the ego-network around a person**
  - **Friend, friends of friends, etc.**
  - **Useful in some cases to avoid catastrophic complexity**

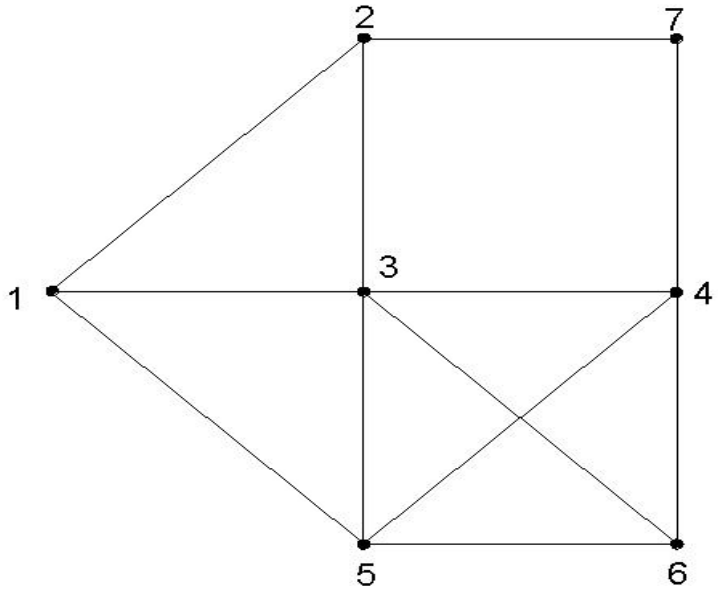


Questions in SNA (2)
What are the groups an actor is in?

- **Find all the fully connected graphs around a person (a clique)**
  - **family-clique, a work-clique, a golf-clique, etc..**



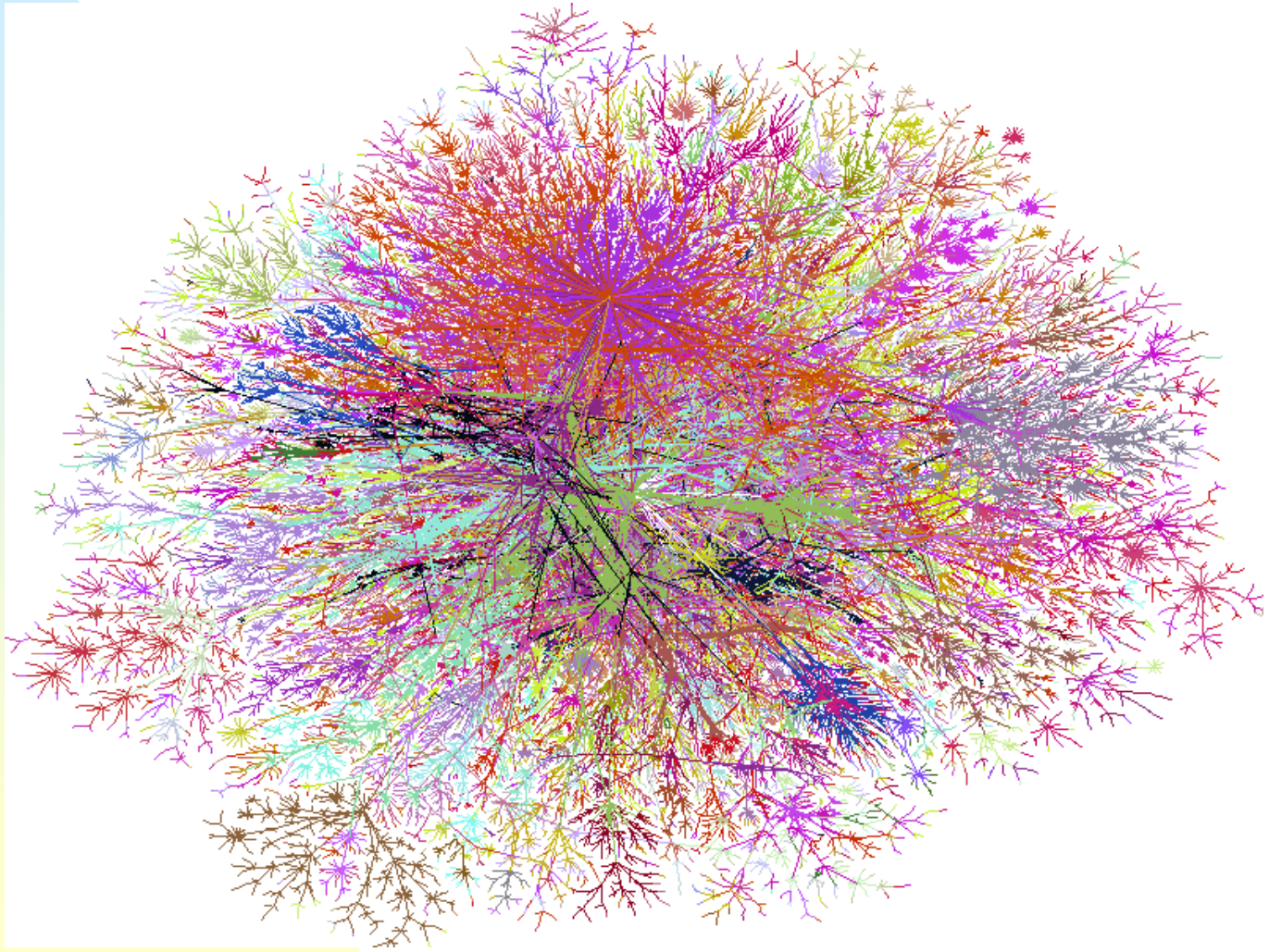
7.2 Subgroups Based on Complete Mutuality



cliques: {1,2,3}, {1,3,5}, and {3,4,5,6}

Fig. 7.1. A graph and its cliques

Questions in SNA (3)
Who are the key players in a network?

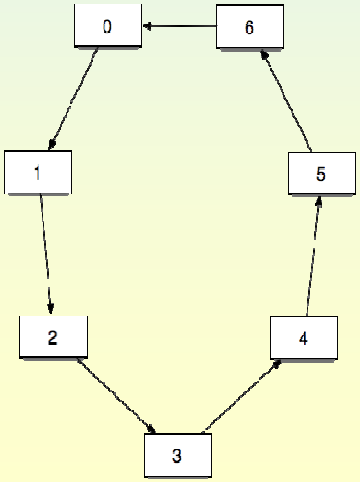


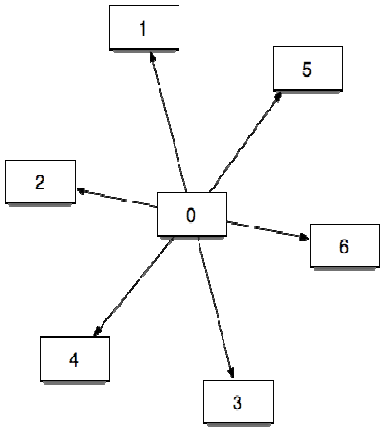
Questions in SNA (3 cont)
How Important is an actor?

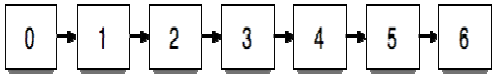
- In-degree, out-degree
- Actor degree centrality
  - I have the most connections in a group so I am more important
- Actor closeness centrality
  - I have more shortest paths to anyone else in the group so I am more important
- Actor betweenness centrality
  - I am more often on the shortest path between other people in the group so I am more important. I can control flow of information better than other people

Actor-degree-centrality (actor, subgroup, generator)

- **Examples**
  - **circle: all degrees are the same**
  - **star: high for the middle, low for the ends**
  - **line: depends where you are in the line, lower at the end**

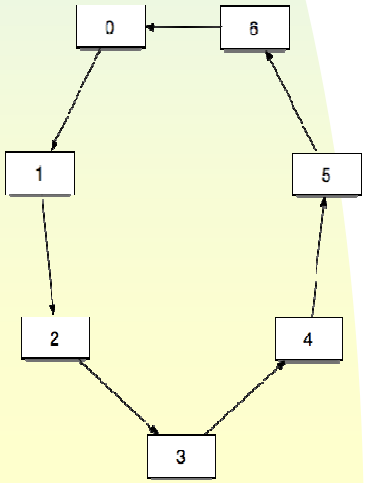


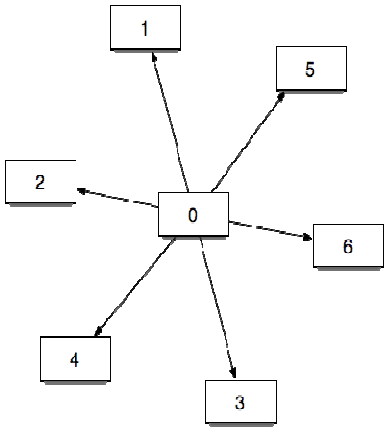


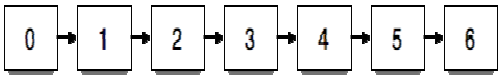


Actor-closeness-centrality (actor, subgroup, generator)

- The (normalized) inverse average path length of all the shortest paths between the actor and every other member of the group. (Inverse so that higher values indicate more central actors).
- Examples
  - star: highest for the middle
  - line: a little bit higher in the middle
  - circle: all the same

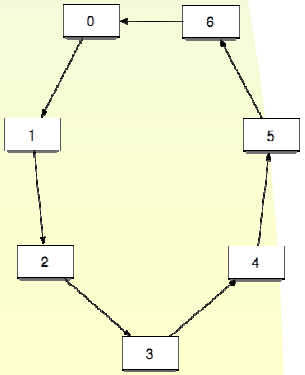


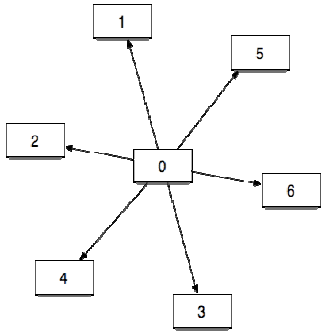


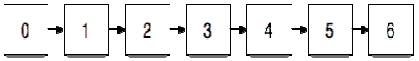


actor-betweenness-centrality (actor, subgroup, generator)

- The actor-betweenness-centrality of actor i is computed by counting the number of shortest paths between all pairs of actors (not including i) that pass through actor i. The assumption being that this is the chance that actor i can control the interaction between j and k
- Example
  - star: the middle one is super important (1)
  - line: lower (at the end are zero)
  - circle: every one the same again

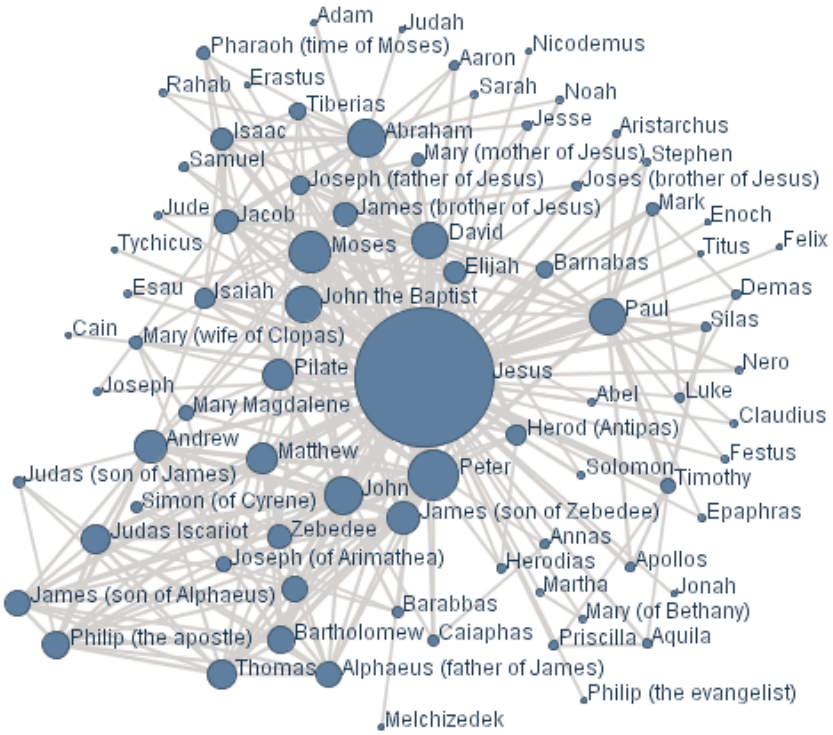






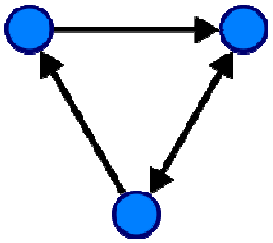
Questions in SNA (4) : has the group a leader, is the group cohesive?

- **Group centralization**
  - How centralized is this group?
  - Does this group have a leader
  - Is there someone controlling the information flow
- **Group cohesiveness**
  - How strong and well connected is this group
  - Are most people connected
  - What is the density

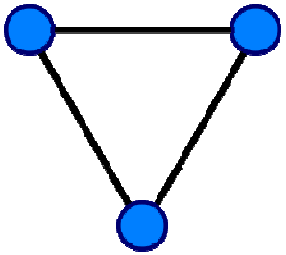


Some technical concepts before we demo (1)

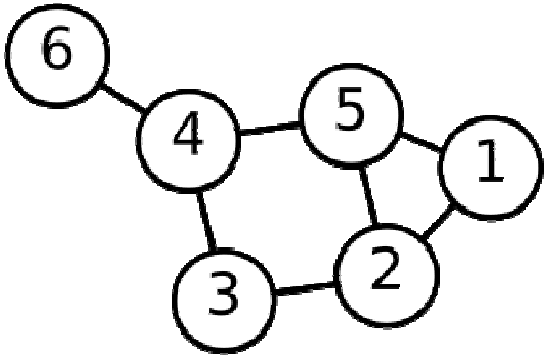
- Directed graph
  - A calls B



- Undirected graphs
  - A knows B &
  - B knows A

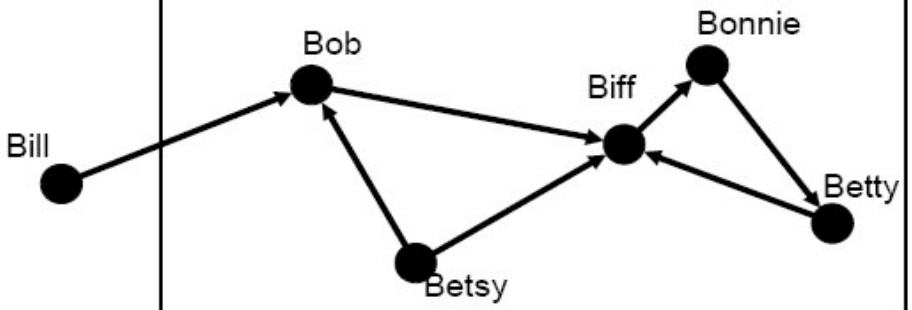
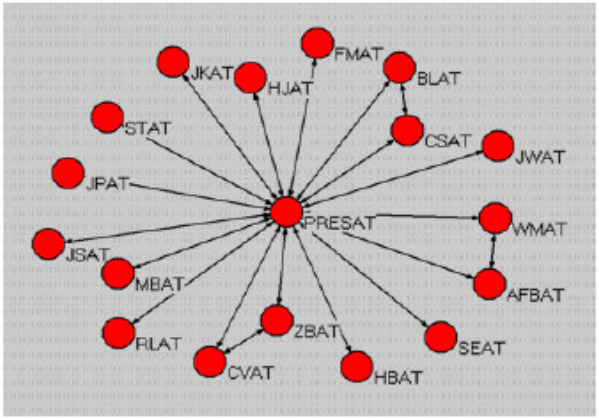
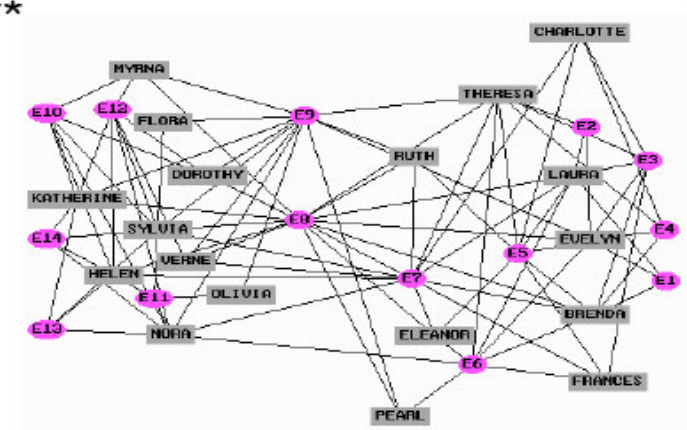
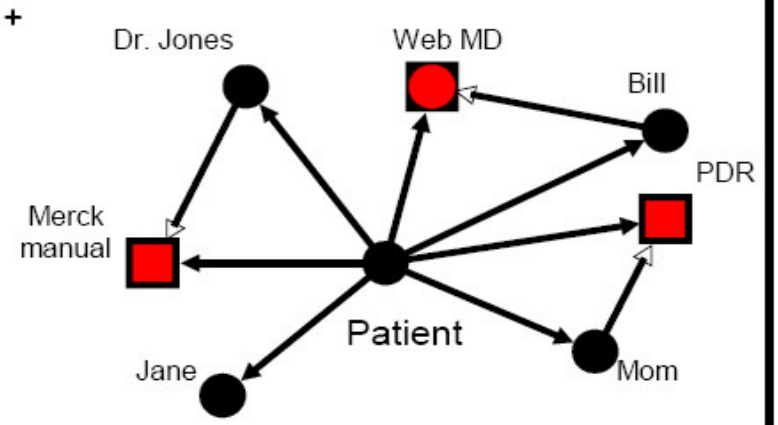


- Most of the graphs we work with are cyclic undirected graphs



FRANZ INC.
Web 3.0's Database

Some technical concepts before we demo (2)

	Complete	Ego
1-mode	++++ 	++++ 
2-mode	** 	+ 

From: http://www.analytictech.com/essex/schedule.htm

Some technical concepts before we demo (3)

- Generators
  - Functions that know how to expand nodes.
  - Fully functional, can be complex sparql or prolog queries
  - Used by all the search functions and social network analysis functions

How to get from A to E??

subj	pred	obj
a	dinner-with	b
a	kissed-with	c
c	movie-with	e
b	kissed-with	d
d	movie-with	e
e	dinner-with	a

```
(defgenerator knows (node)
   (objects-of :p dinner-with))

(defgenerator knows (node)
   (objects-of :p dinner-with)
   (subjects-of :p dinner-with))
```

How to get from A to E??

```
(defgenerator knows ()
  (object-of  :p dinner-with)
  (subject-of :p dinner-with)
  (object-of  :p movie-with)
  (subject-of :p movie-with)
  (object-of  :p kissed-with)
  (subject-of :p kissed-with))

(defgenerator knows ()
  (undirected (dinner-with movie-with kissed-with)))
```

How to get from A to E

```
(defgenerator knows (node)
   (select (?x)
      (q- (?? node) movie-with ?x)
      (q- (?? node) dinner-with ?x)
      (not (q- node kissed-with ?x)))
    (select (?x)
      (q- ?x movie-with (?? node))
      (q- ?x dinner-with (?? node))
      (not (q- ?x kissed-with (?? node)))))
```

General search functions

(bidirectional-search a b generator exhaustive depth ?p)

- A and B are begin and end node
- Generator: any function that takes a node as input and returns a list of nodes
- Exhaustive: if true: find all paths
- Depth: stop when depths is more than….

Sample SNA functions

Actor = single node

Group = list of nodes

Depth = number

Generator = generator

(Ego-group actor generator depth ?group)

- binds ?group to group of nodes

(Cliques actor generator min-depth ?cl)

- binds ?cl to all cliques

(Clique-members actor generator min-depth ?cl ?a)

- binds ?cl to cliques and then iterates of ever member ?a in ?cl

(Actor-centrality actor group generator ?num)

- binds ?num to actorcentrality

(Actor-centrality-members group ?actor ?num)

- binds ?actor to every actor in group, ?centrality is centrality of

that actor, we start with the actor with highest centrality.

(Group-centrality group generator ?num)

Integrated in select language.

```
(defgenerator knows (node)
   (undirected :p (!fr:dinner-with !fr:kissed-with)))

(select (?x)
   (ego-group-members !person:jans knows ?x 2)
   (q ?x !geo:place ?y)
   (geo-box-around !geoname:Berkeley ?y 5 miles))

(select (?x)
   (ego-group !person:jans knows ?group 2)
   (actor-centrality-members ?group knows ?x ?num)
   (q ?x !geo:place ?y)
   (geo-box-around !geoname:Berkeley ?y 5 miles)
   !)
```

Why is this not in SPARQL

- We would love to
  - > 2 arguments is currently problem
  - Most predicates would have to be magic
  - I want some more feedback from the community

Demo [1] : a database of friendship and love …

- 5000 people in a database

Each person knows on average 40 others

Shook-hands-with with 35

Restaurant-with 25

Movie-with 20

Kissed-with 10

Intimate-with 4

Demo [2] DBpedia

- DBpedia is RDF version of wikipedia
- Many thanks to people at www.dbpedia.org
- Current version is 83.000.000 triples (without page links)
- We added
  - Sparql & Select.
  - Text indexing
  - A SNA movie demo demonstrating some SNA

And then…

- Use AllegroGraph for scalable graph applications
- New opportunities with SNA libraries
- Please: if you have SNA/graph algorithms you need, we will help you implement them.
- Next month:
  - Ego-groups as cached datastructures
  - Eigen Vector Centrality (high correlation with centrality measures but infinitely faster)
  - Pagerank as used by Google