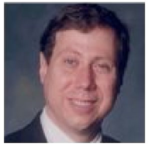
The Right Way to Search Evolving Graphs

Jiahao Chen & Weijian Zhang

The University of Manchester

GABB 2016, Chicago
05-23

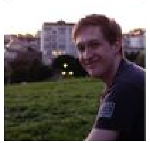
The Julia Lab at MIT


Alan Edelman


Jiahao Chen


Andreas Noack


Xianyi Zhang

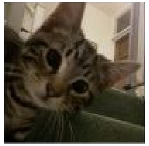

Jarrett Revels


Eka Palamadai


David Sanders


Oscar Blumberg


Isaac Virshup


Simon Danisch


Amit Murthy


Tanmay Mohapatra


Shashi Gowda


Jake Bolewski
USAP


Weijian Zhang
U. Manchester

Collaborators


Yee Sian Ng


Joey Huchette


Miles Lubin


Iain Dunning


Steven Johnson
MIT Mathematics


Alex Townsend
MIT Mathematics


Yichao Yu
Harvard


Pete Szolovits
CSAIL


Jeremy Kepner
Lincoln Labs


Stavros Papadopoulos
Intel Labs


Nikos Patsopoulos
Brigham Woman's Hospital


Jon Malmaud


Simon Kornblith


Jack Poulson
Stanford

Saudi Aramco

SMART
Singapore-MIT Alliance for Research and Technology

NSF

ISTC BIG DATA

XDATA

DARPA

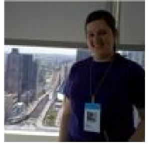
NIH

NIH Big Data to Knowledge (BD2K)

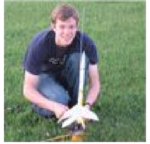
bigdata@CSAIL
MIT BIG DATA INITIATIVE

Summer of Code alums 2013

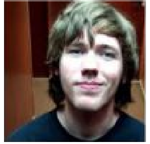
Keno Fischer
Julia Computing



Leah Hanson
Stripe



Jameson Nash
Julia Computing



John Myles White
Facebook

2014

Simon Danisch
Julia Lab

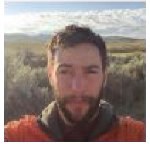
Shashi Gowda
Julia Lab

Mike Innes
Julia Computing

Google SUMMER of CODE TENTH YEAR 2014

2015

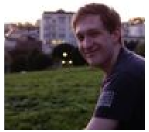
Simon Danish
Julia Lab



David Gold
U. Washington

Jacob Quinn
Domo

NUMFOCUS
OPEN CODE = BETTER SCIENCE



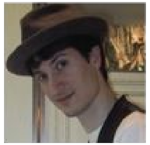
Jarrett Revels
Julia Lab

Kenta Sato
U. Tokyo

Rohit Varkey Thankachan
Nat'l Inst. Tech. Karnataka

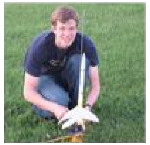
GORDON AND BETTY MOORE FOUNDATION

Alums at Julia computing



Jeff Bezanson

Stefan Karpinski

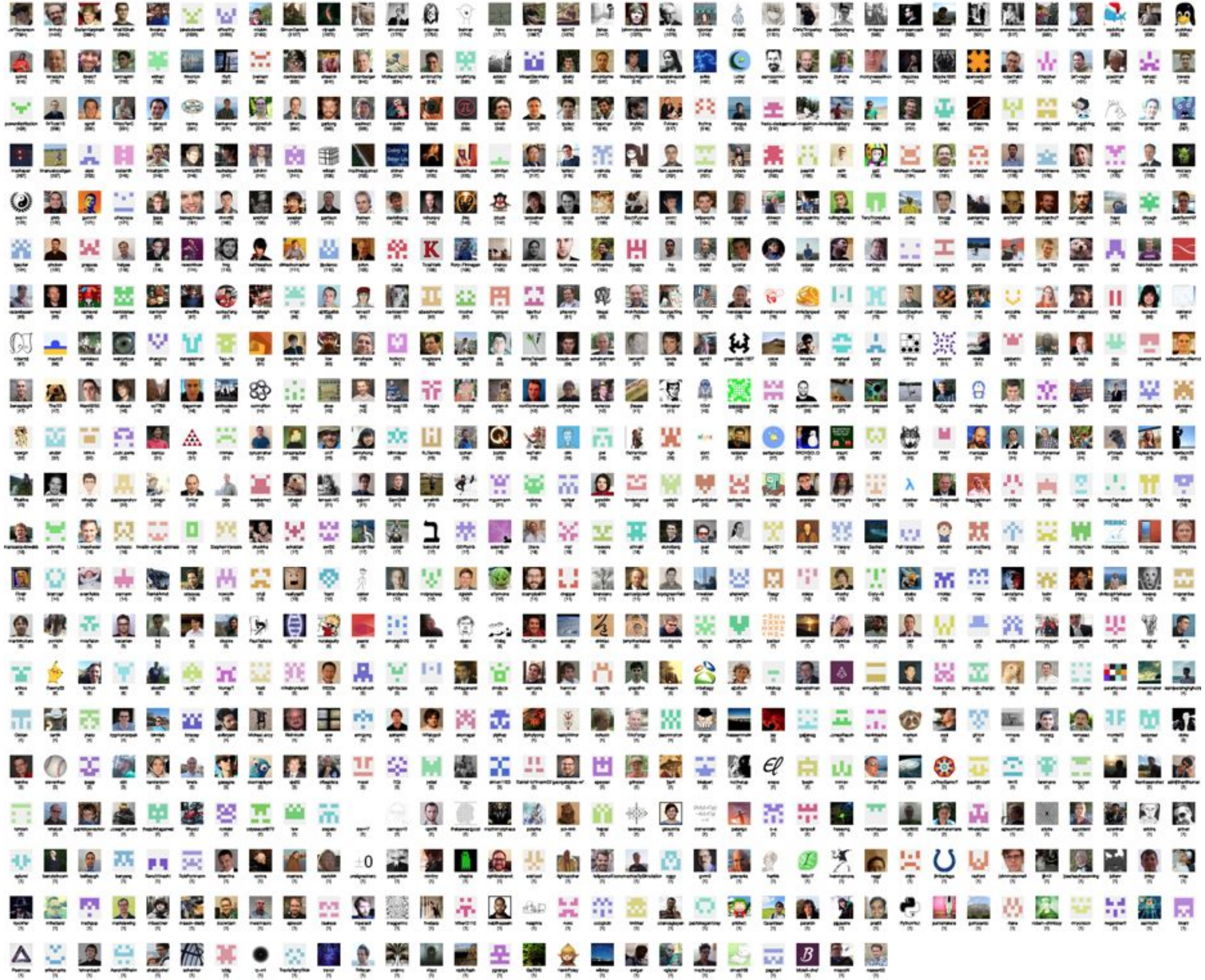


Jameson Nash

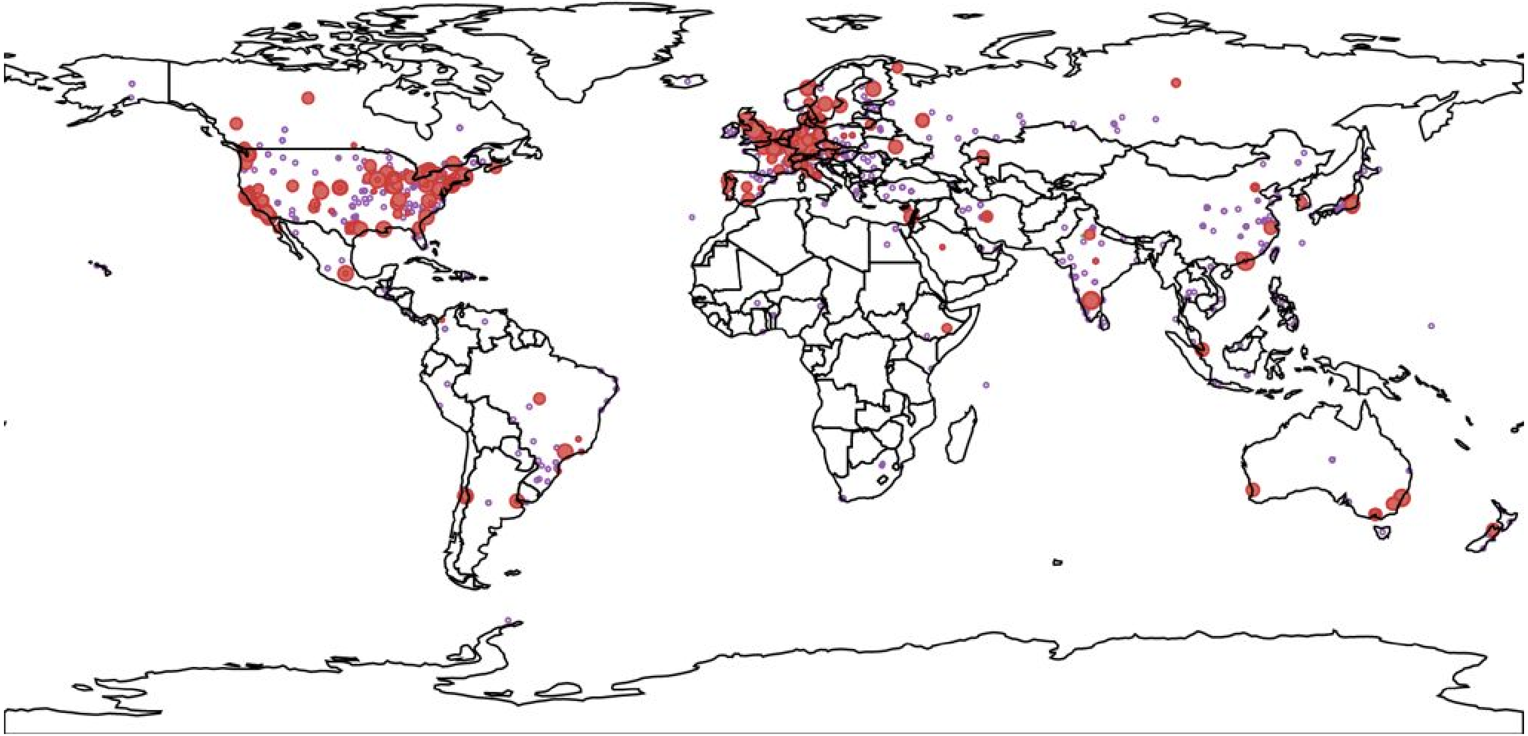
Keno Fischer

Viral B. Shah

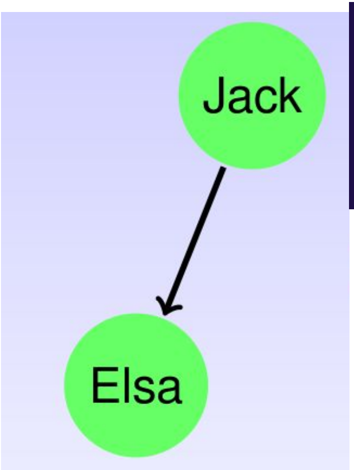
Mike Innes

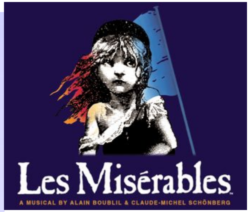


The world of julia

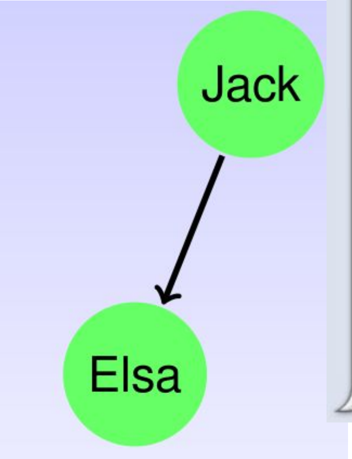


https://github.com/jiahao/ijulia-notebooks

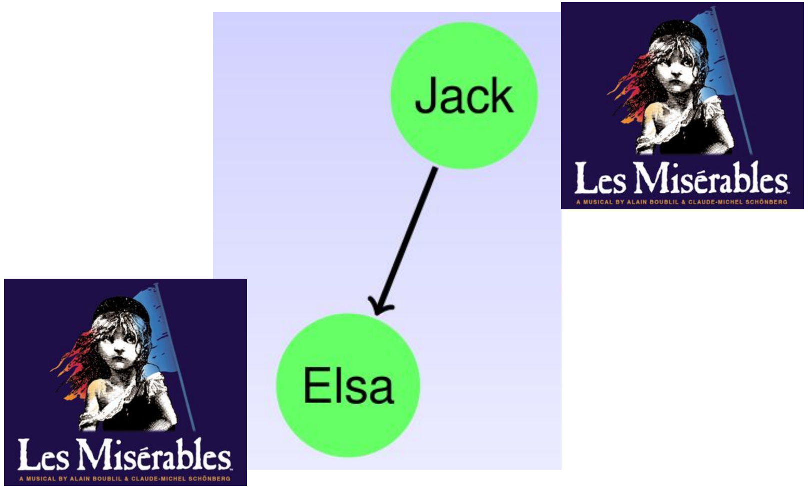




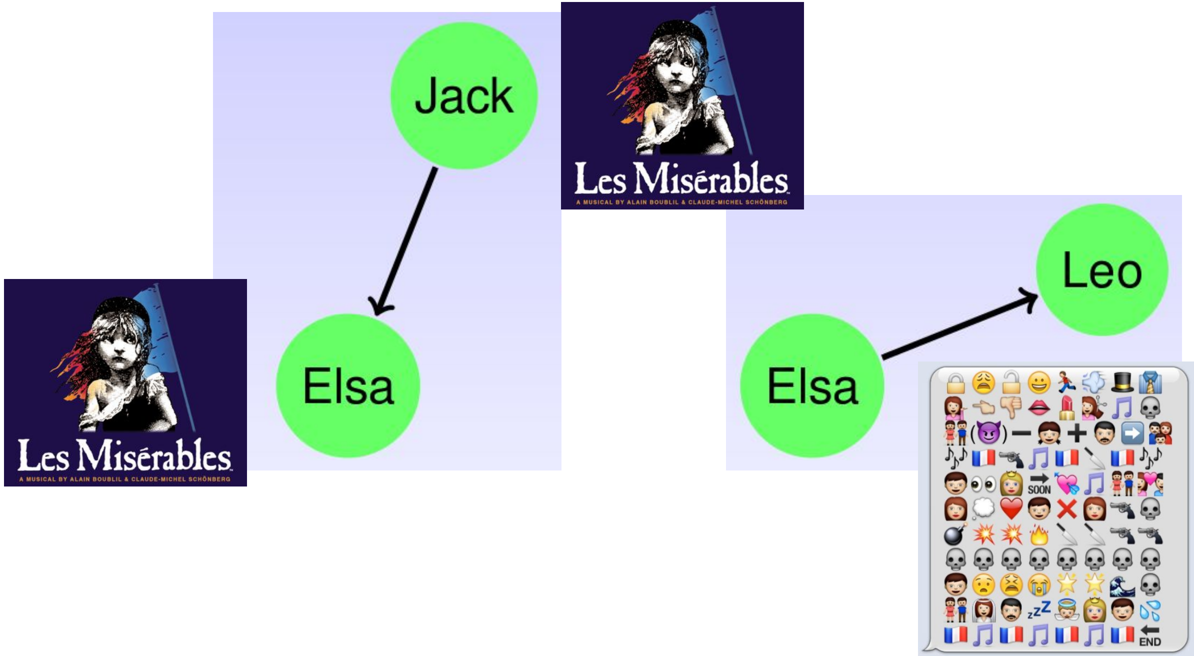
Jack likes Les Mis.



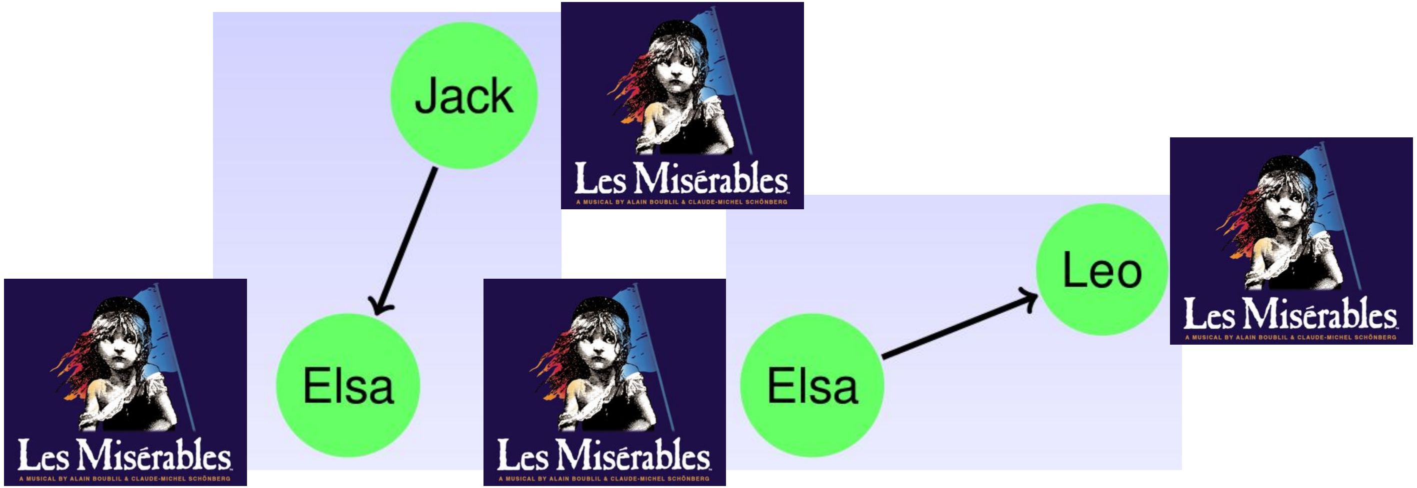
Jack tells Elsa about Les Mis



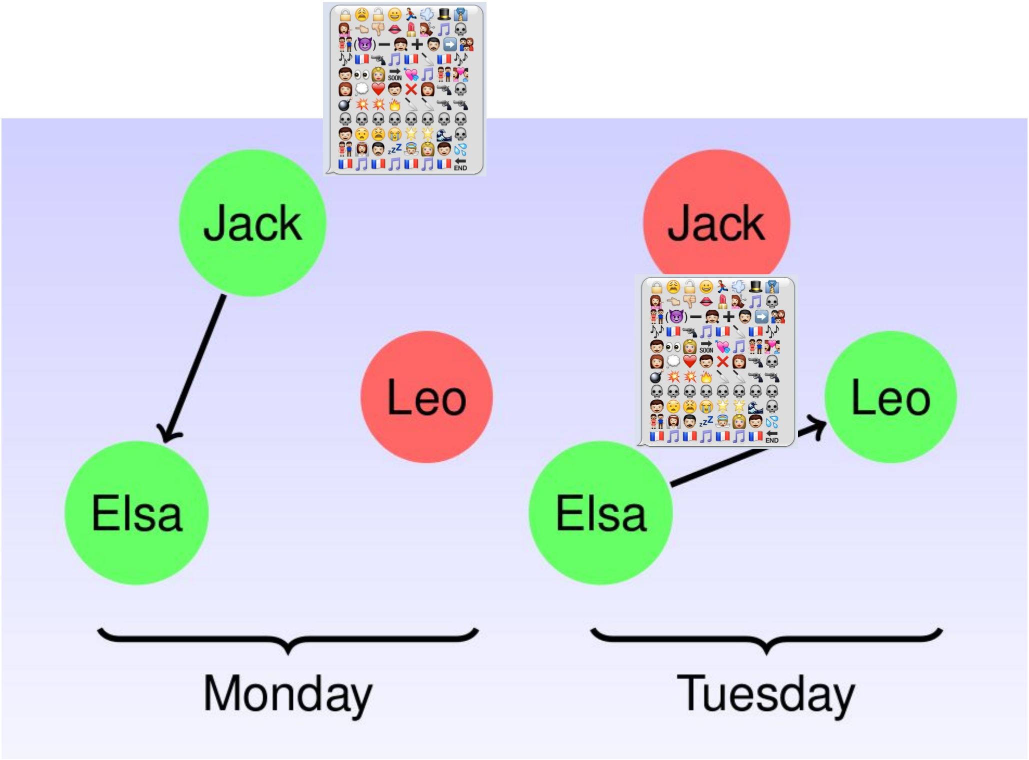
Both Jack and Elsa now like Les Mis.



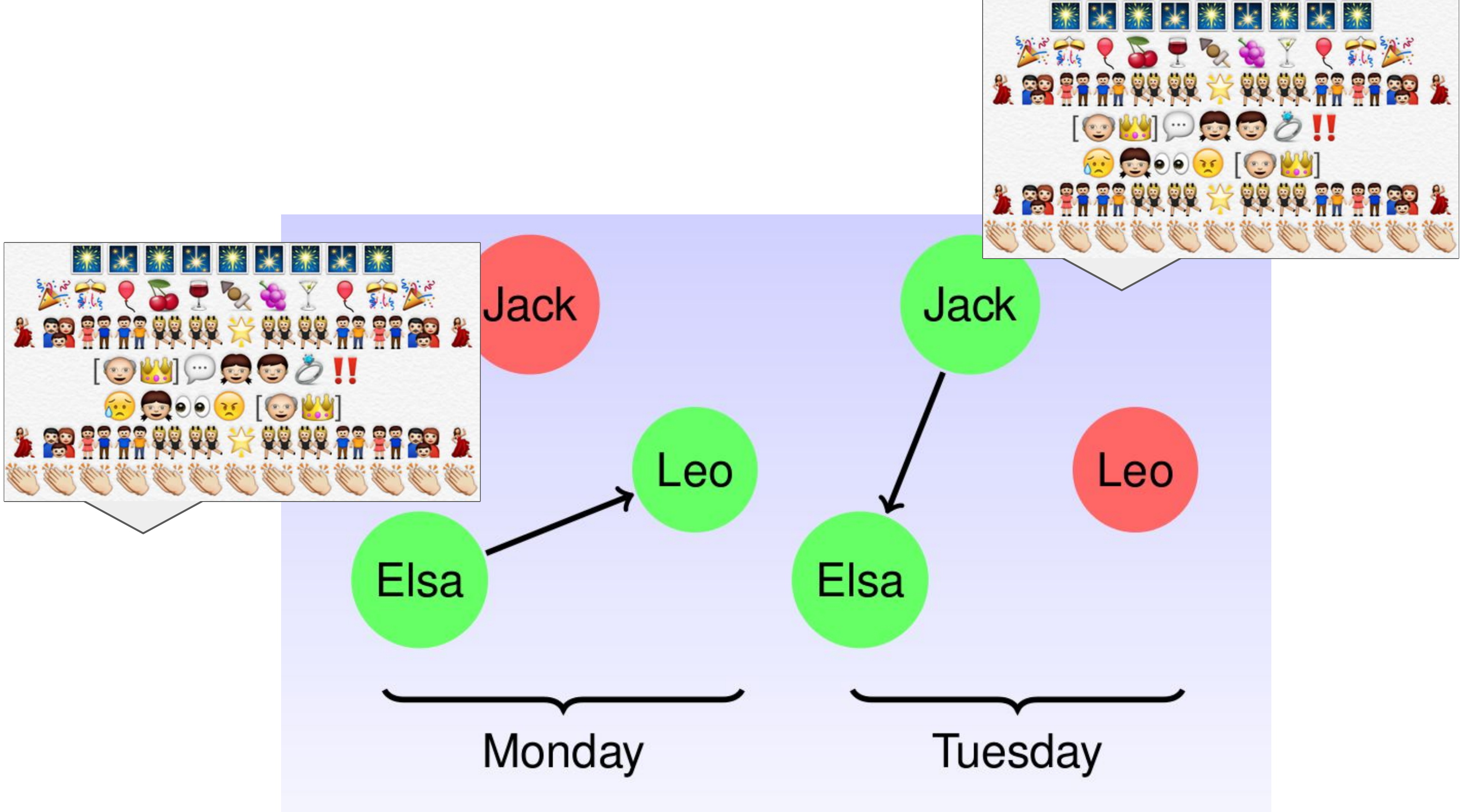
Elsa tells Leo about Les Mis.



Now everyone likes Les Mis.

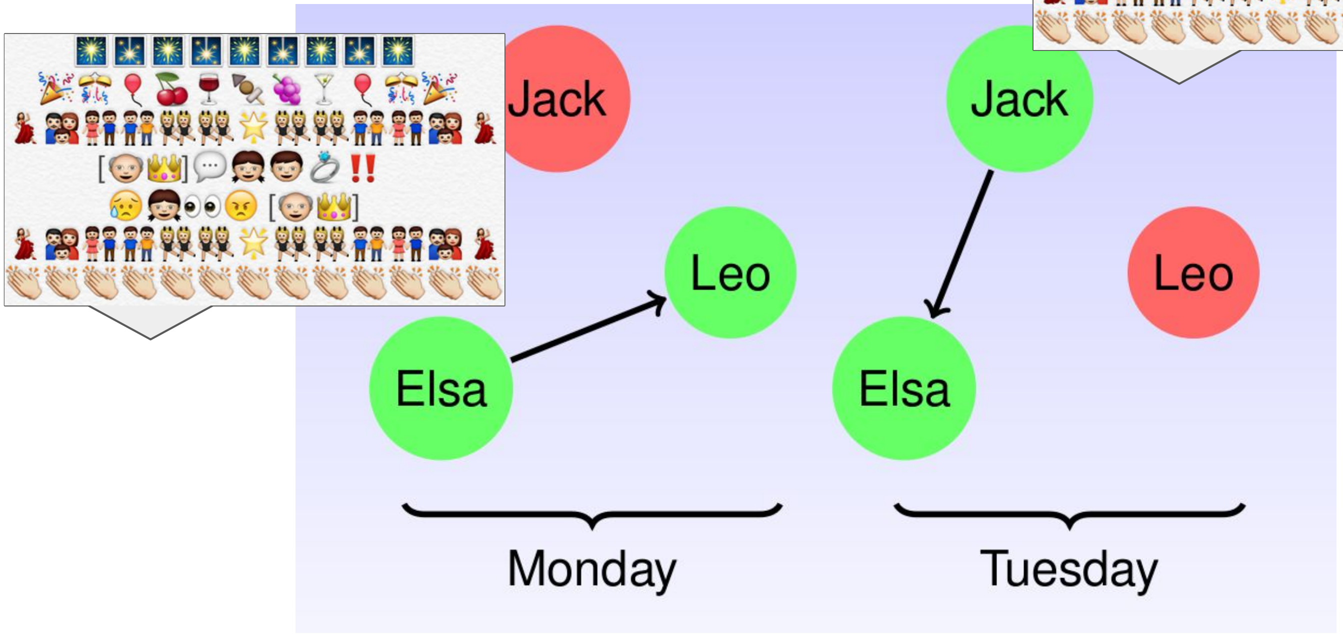


Leo heard about Les Mis from Jack.



Leo did not hear about The Hunger Games from Jack.

Information flow on a network is causal

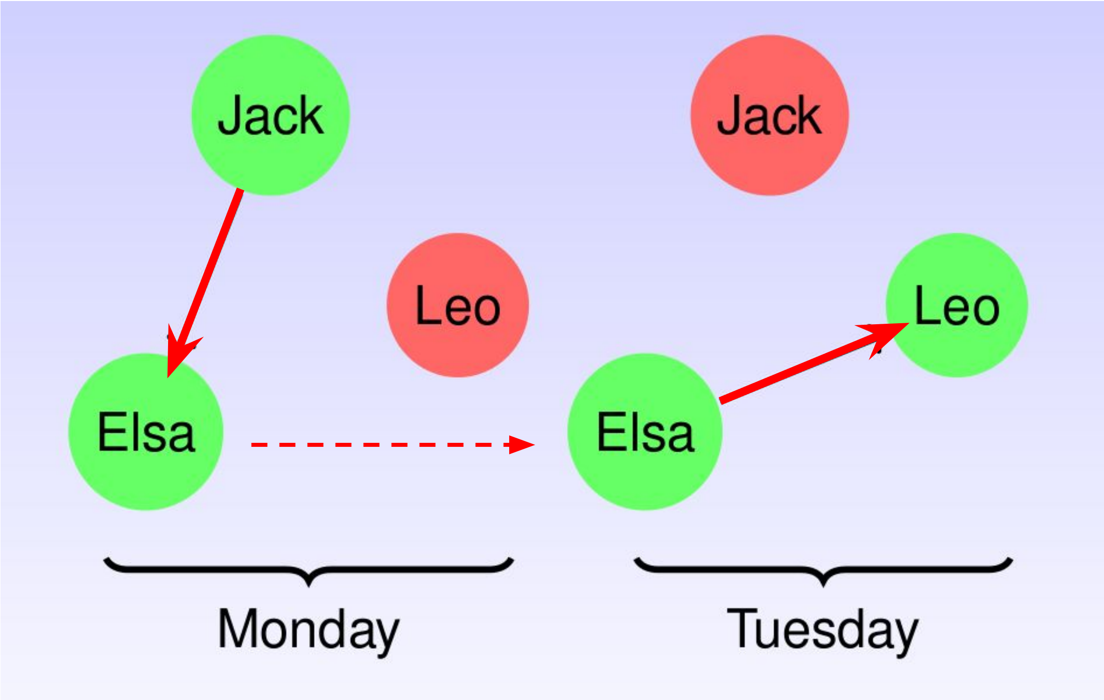


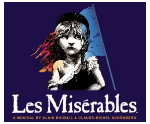
Leo did not hear about The Hunger Games from Jack.

Information flow on a network is causal

Unless you have one of these...

Temporal paths: causal connections in space and time

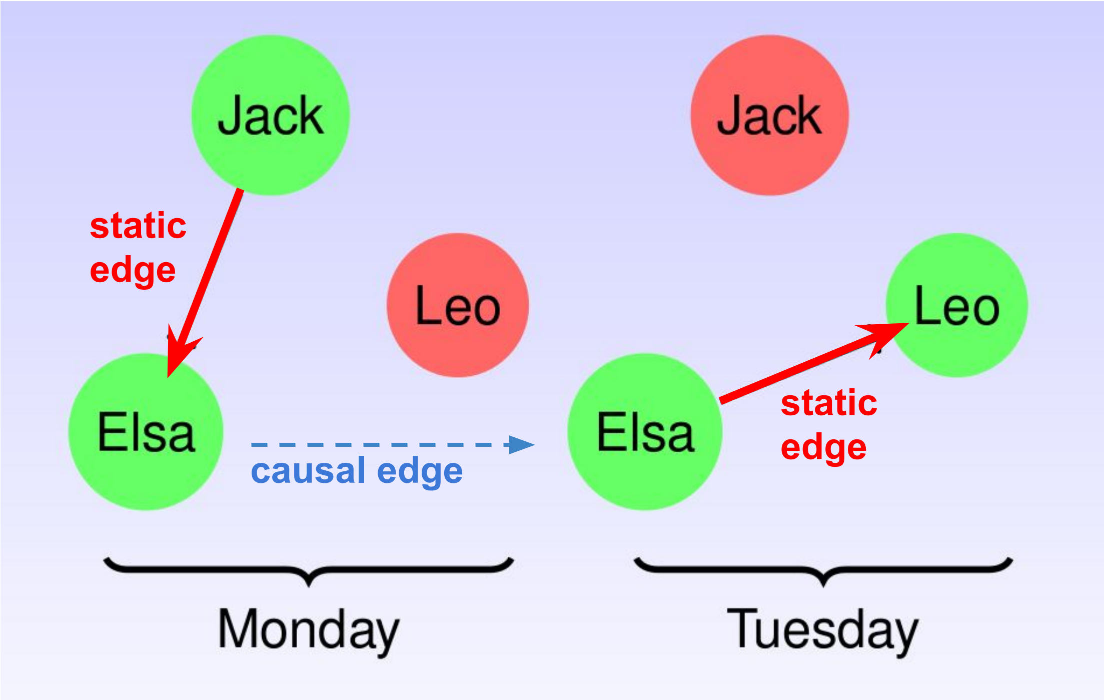




(Jack, Mon) → (Elsa, Mon) → (Elsa, Tues) → (Leo, Tues)

In general, not trivial to find all temporal paths

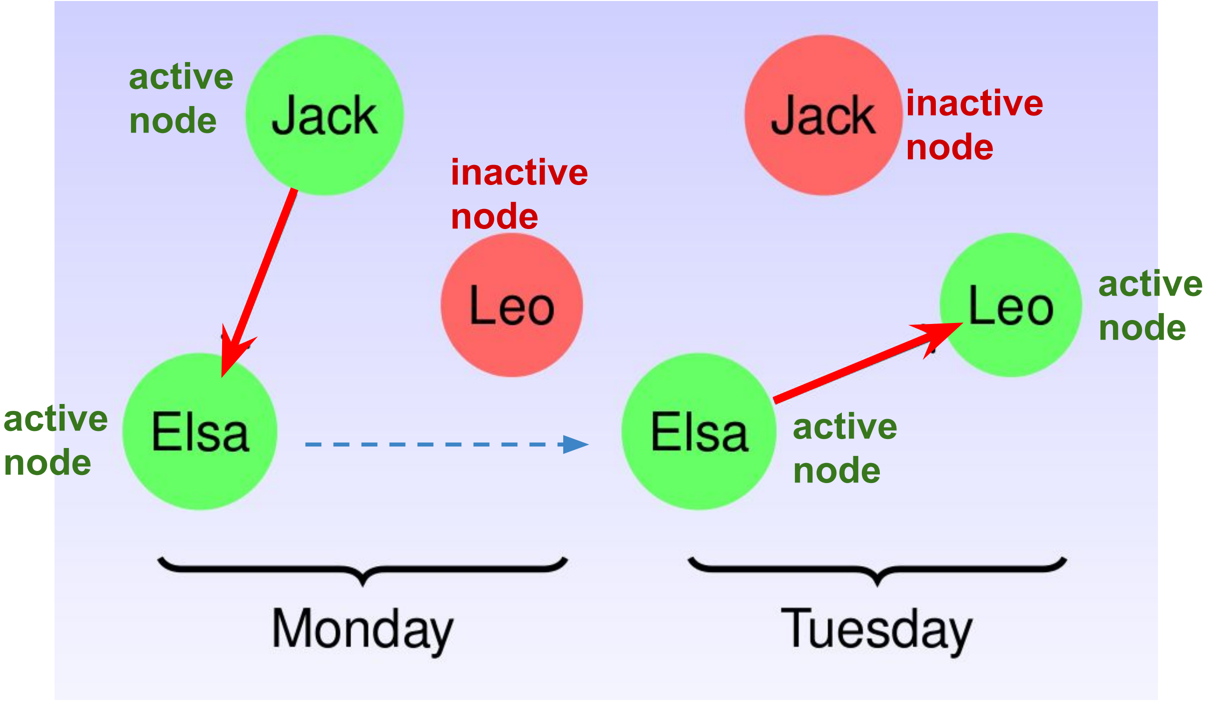
Temporal paths: causal connections in space and time



Grindrod et al. (2011) only count static edges

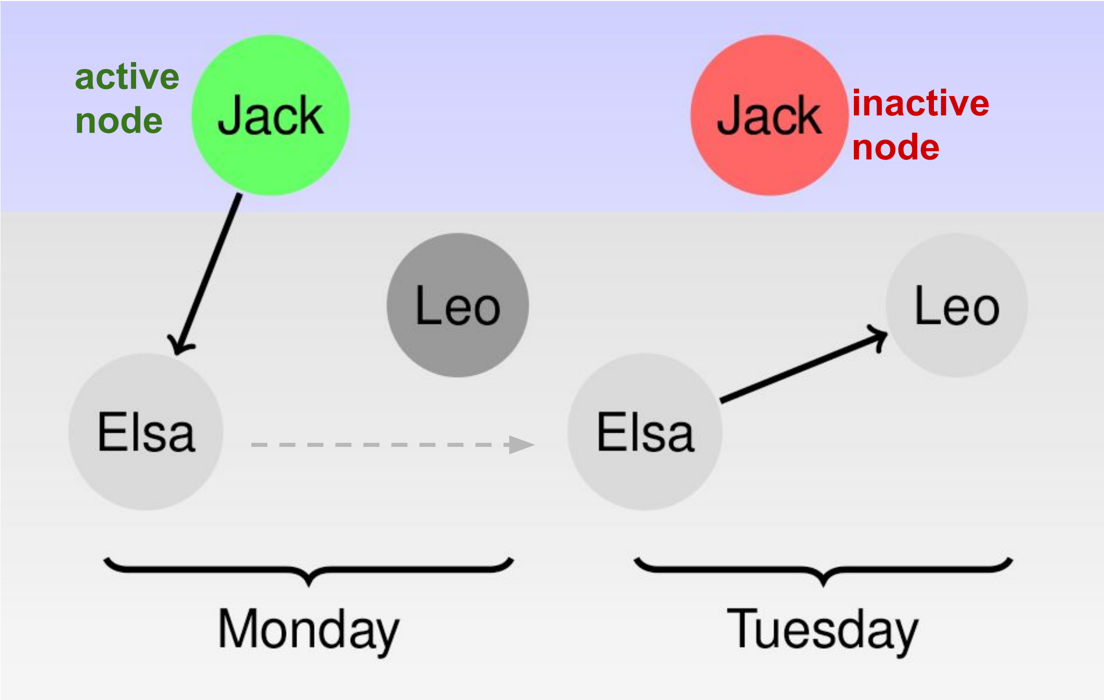
Tang, Musolesi, and Mascolo (2009) only count causal edges

Active and inactive nodes



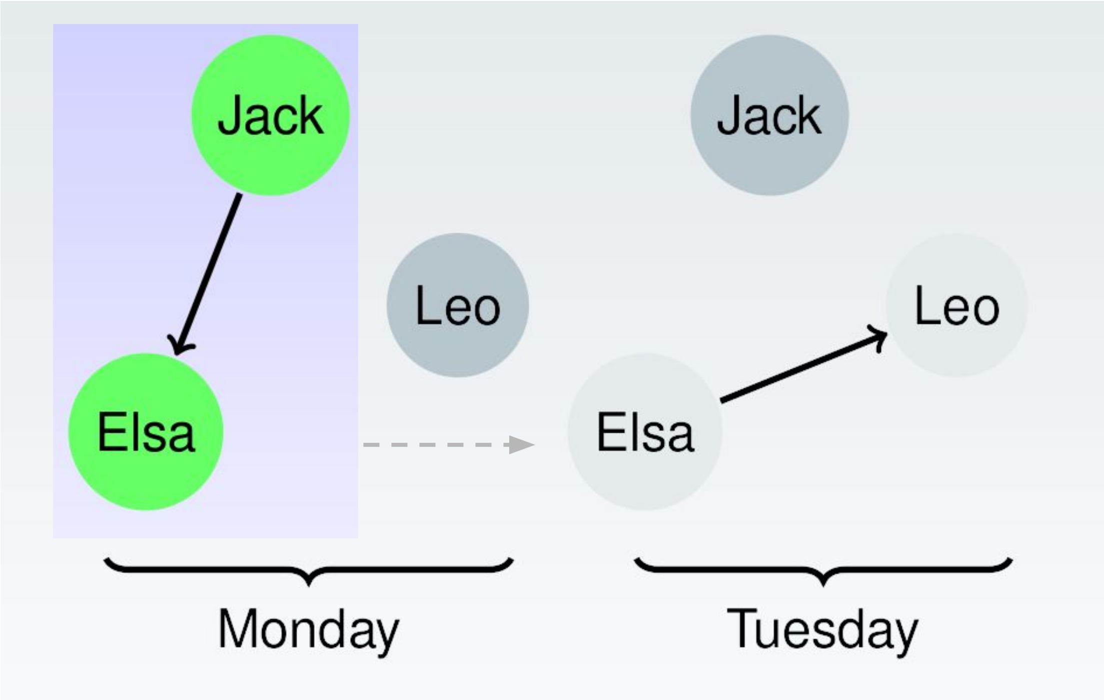
Temporal paths connect active nodes only

Active and inactive nodes



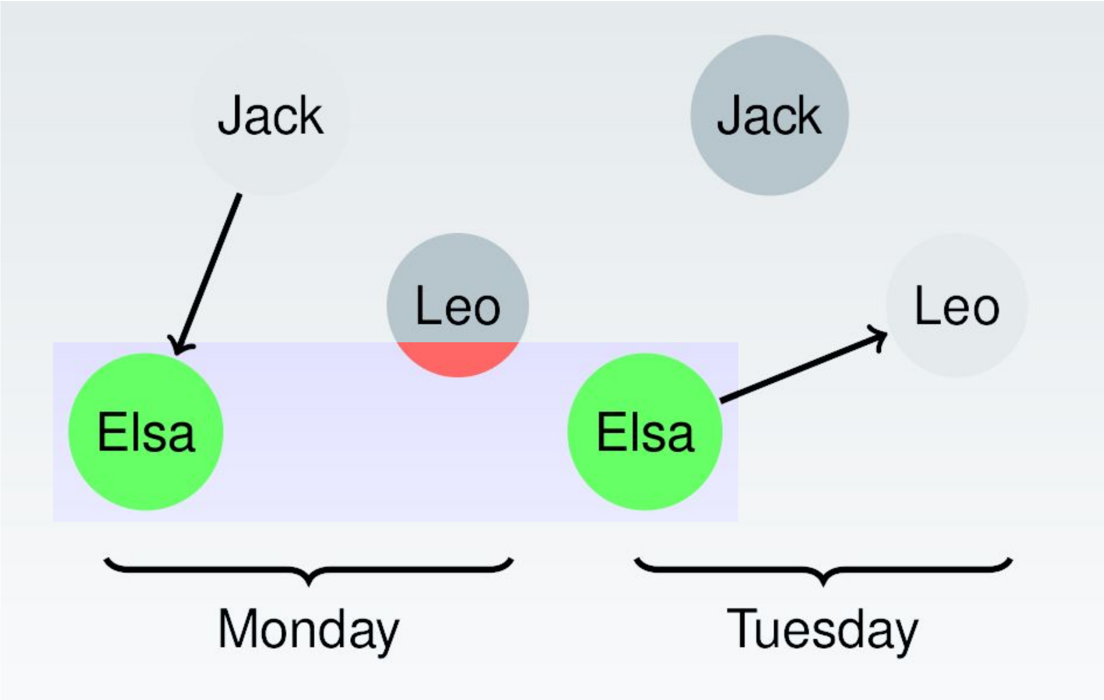
The same node may be active or inactive at different times

Forward neighbors



The **forward neighbor** of (Jack, Monday) is (Elsa, Monday)

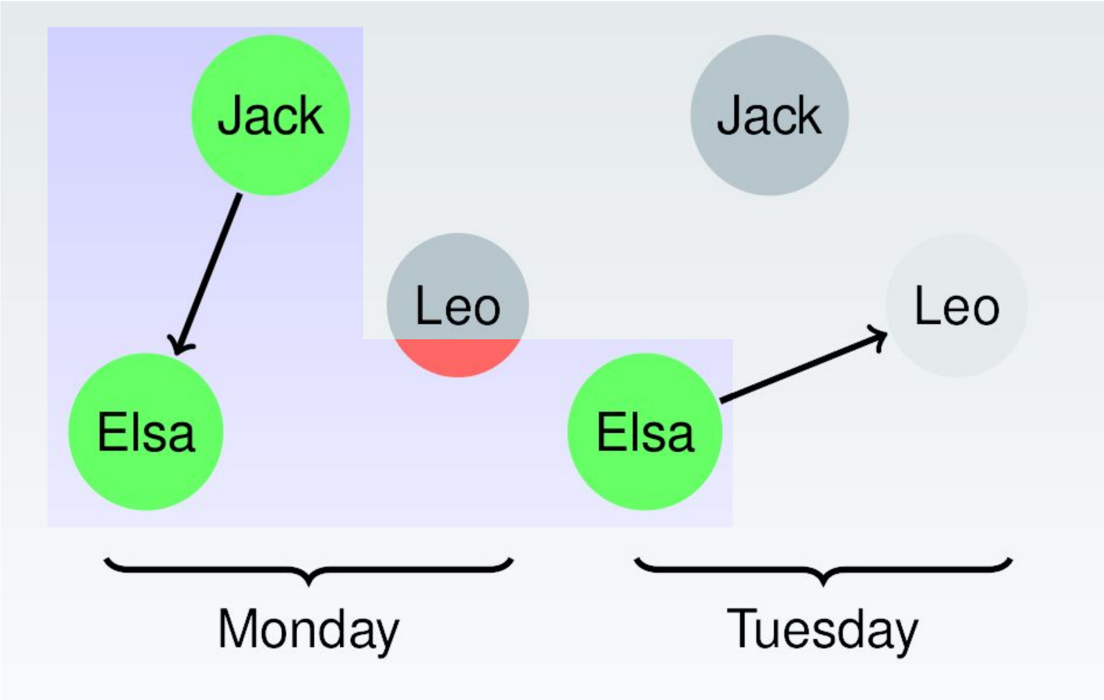
Forward neighbors



The **forward neighbor** of (Jack, Monday) is (Elsa, Monday)

The **forward neighbor** of (Elsa, Monday) is (Elsa, Tuesday)

Forward neighbors

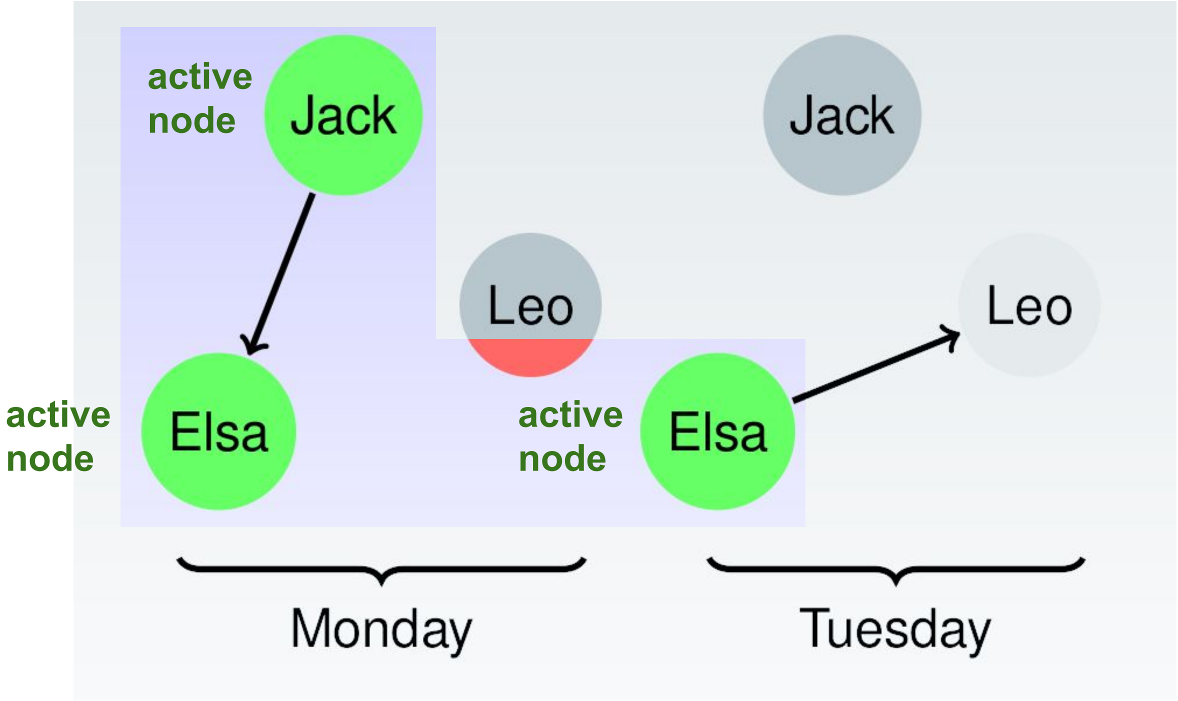


The **forward neighbor** of (Jack, Monday) is (Elsa, Monday)

The **forward neighbor** of (Elsa, Monday) is (Elsa, Tuesday)

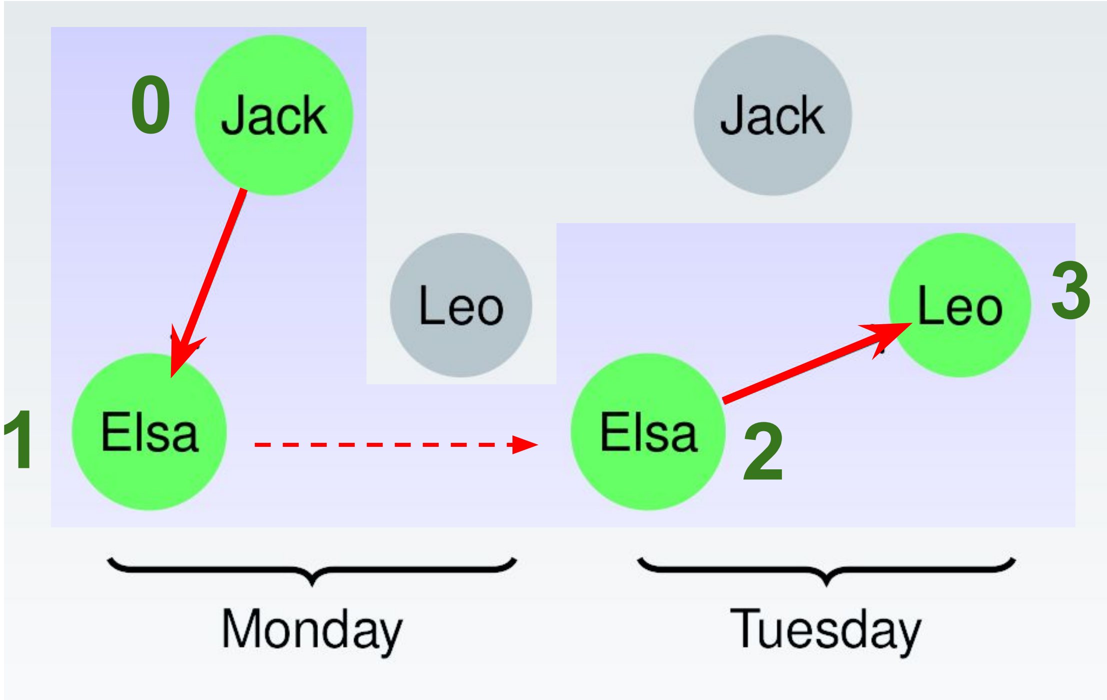
(Elsa, Tuesday) is **reachable** from (Jack, Monday)

Forward neighbors

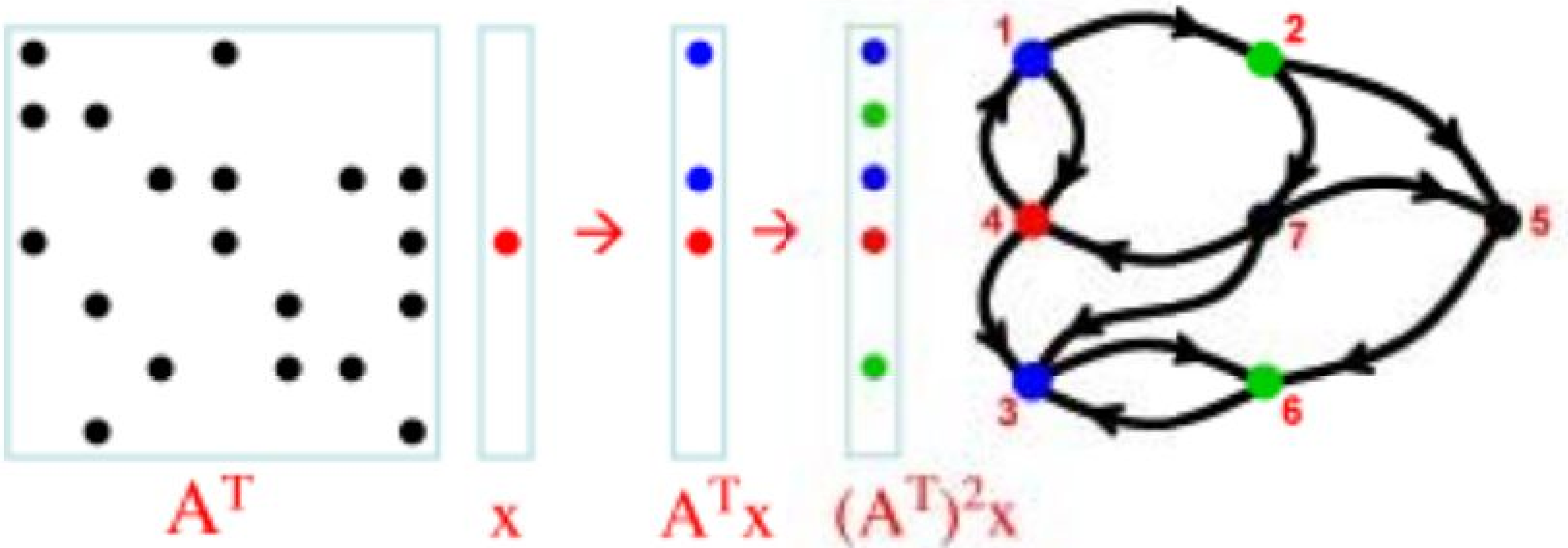


All **forward neighbors** are **active nodes**

Breadth-first search finds forward neighbors

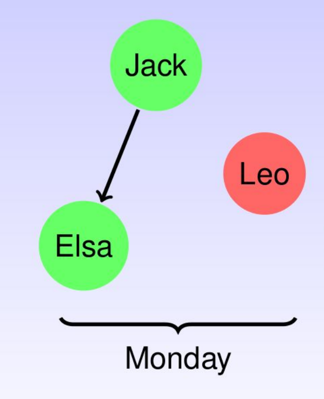


Breadth-first search as a matrix-vector product

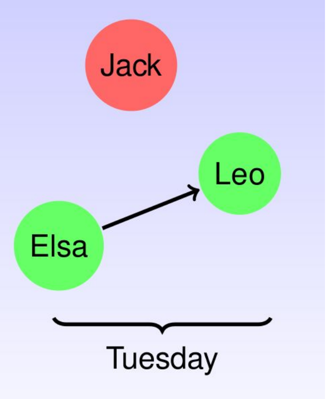


“High-performance graph algorithms from parallel sparse matrices”, John R Gilbert, Steve Reinhardt, Viral B Shah, 2006, *Applied Parallel Computing: State of the Art in Scientific Computing*, Lecture Notes in Computer Science vol. 4699, pp 260-269. [doi: 10.1007/978-3-540-75755-9_32](https://doi.org/10.1007/978-3-540-75755-9_32)

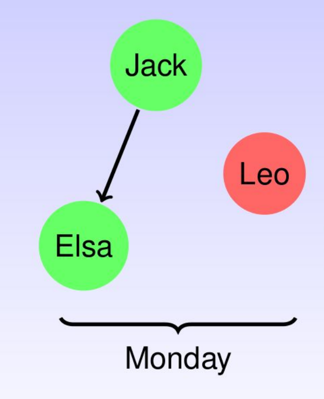
Breadth-first search as a matrix-vector product



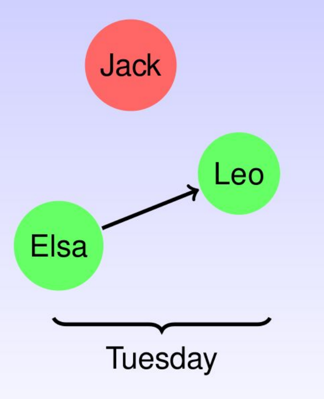
$$\begin{array}{ccc} \text{J} & \text{E} & \text{L} \\ \end{array}$$
$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$



Breadth-first search as a matrix-vector product

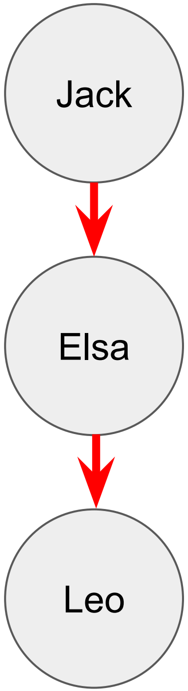


$$\begin{array}{c} \begin{array}{ccc} J & E & L \end{array} \\ \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{Monday}} \end{array} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{Tuesday}}$$



Where does the temporal edge go?

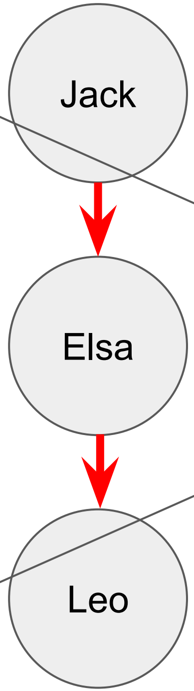
Static graph correspondence



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

Problem: does not encode time ordering of edges

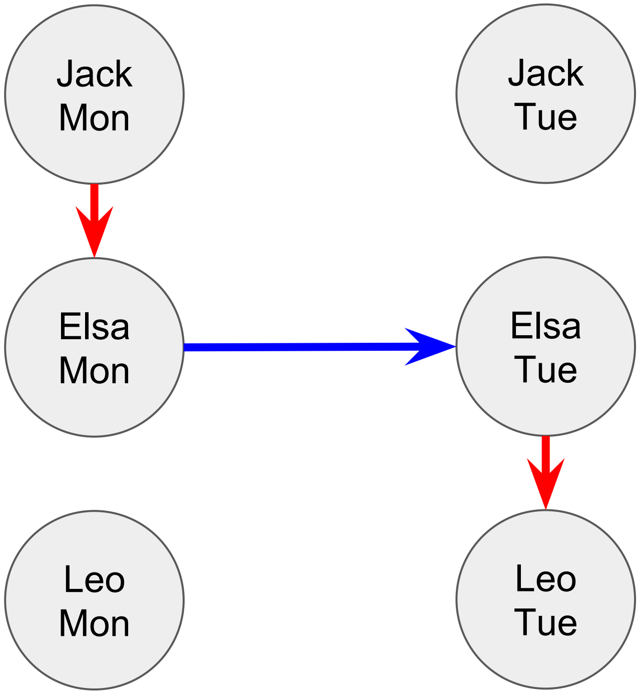
Static graph correspondence



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

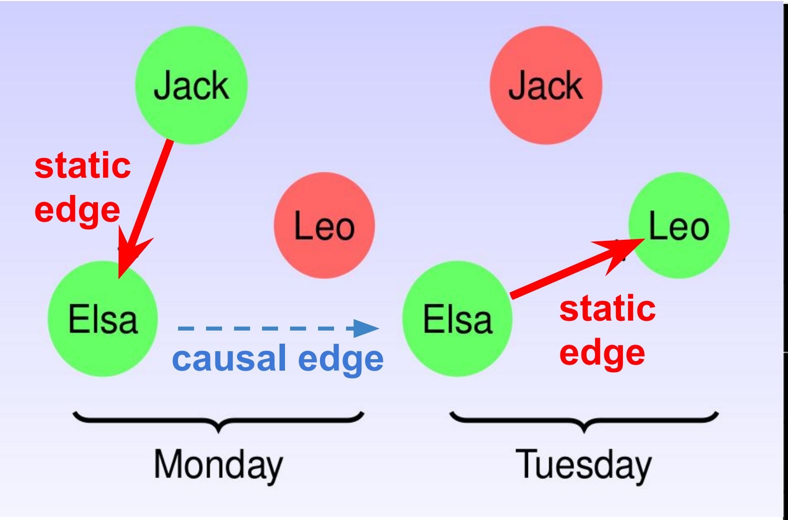
Problem: does not encode time ordering of edges

Static graph correspondence



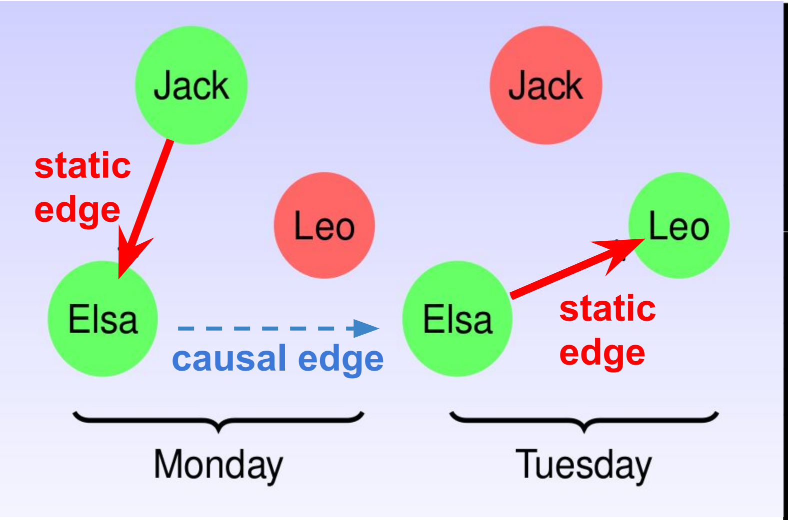
$$
\begin{array}{c}
\begin{array}{cccccc} (J, Mon) & (E, Mon) & (L, Mon) & (J, Tue) & (E, Tue) & (L, Tue) \end{array} \\
\left[\begin{array}{ccc|ccc}
0 & \color{red}{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \color{blue}{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \color{red}{1} \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

Breadth-first search as a matrix-vector product



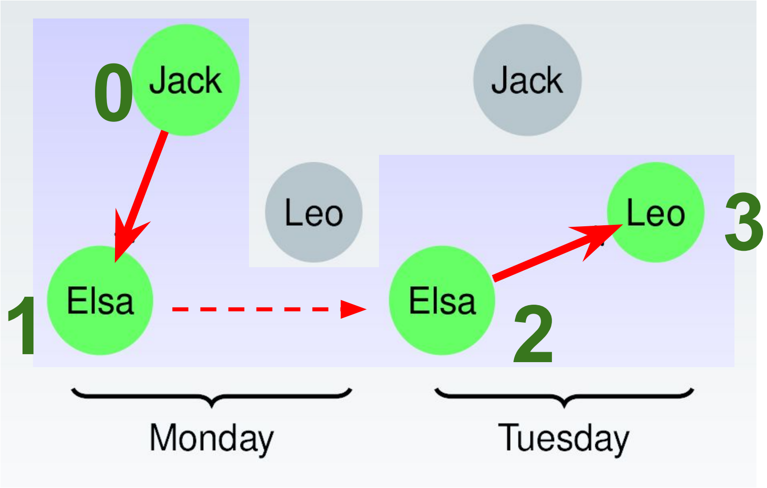
$$
\begin{array}{c}
\begin{array}{cc} \text{Monday} & \text{Tuesday} \end{array} \\
\left[\begin{array}{ccc|ccc}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

Breadth-first search as a matrix-vector product



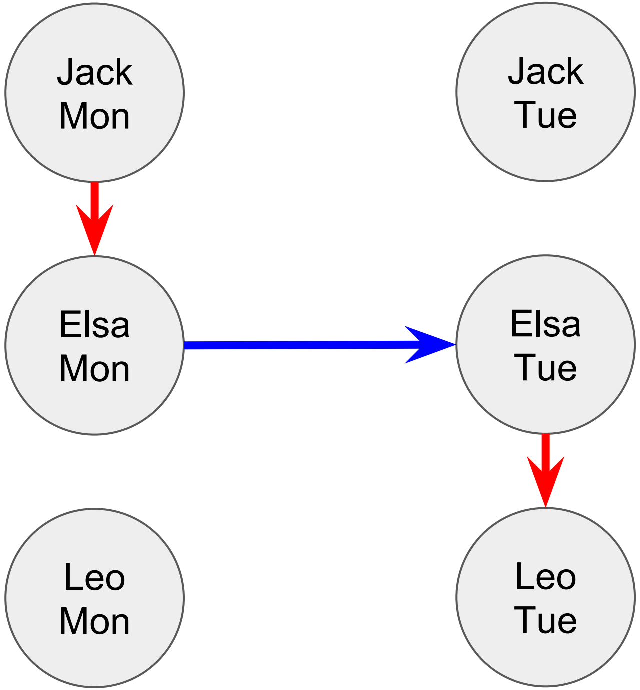
$$\begin{array}{c} \begin{array}{cc} \text{Monday} & \text{Tuesday} \end{array} \\ \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ A \end{array} \begin{array}{c} \\ \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{array}\right] \\ x \end{array}$$

Breadth-first search as a matrix-vector product



$$
\underset{x}{\left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{array}\right]}
\quad
\underset{A^T x}{\left[\begin{array}{c} 0 \\ 1 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{array}\right]}
\quad
\underset{(A^T)^2 x}{\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \end{array}\right]}
\quad
\underset{(A^T)^3 x}{\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 1 \end{array}\right]}
$$

Static graph correspondence



$$\begin{array}{c}
\begin{array}{cccccc}
\text{(J, Mon)} & \text{(E, Mon)} & \text{(L, Mon)} & \text{(J, Tue)} & \text{(E, Tue)} & \text{(L, Tue)}
\end{array} \\
\left[\begin{array}{ccc|ccc}
0 & \color{red}{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \color{blue}{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \color{red}{1} \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}$$

Problem: adjacency matrix of corresponding static graph is big

Adjacency matrix has structure!

$$
\begin{array}{cccccc}
(J, Mon) & (E, Mon) & (L, Mon) & (J, Tue) & (E, Tue) & (L, Tue) \\
\end{array}
$$

$$
\left[\begin{array}{ccc|ccc}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
$$

Diagonal projection matrix

Always 0
Can't go back in time!

Can compute matrix-vectors products efficiently

Conclusions

1. Breadth-first search over temporal paths can be expressed as matrix-vector products
2. Evolving graphs correspond to static graphs with special structure in the adjacency matrix, enabling fast matvecs
3. Implemented in EvolvingGraphs.jl in Julia

What’s next?

1. Parallel implementation
2. Shortest temporal path algorithm using semiring algebra
3. Graph centrality algorithms (PageRank, eigenvector centrality)
4. Analyze citation networks

References

- *Evolving graph models*
- **Leskovec, Kleinberg, and Faloutsos,** (2007) Graph Evolving: Densification and Shrinking Diameters
- **Leskovec et al.** (2008) Microscopic Evolution of Social Networks
- *Evolving graph metrics and centralities*
- **Tang, Musolesi, and Mascolo** (2009)Temporal distance metrics for social network analysis
- **Tang et al.** (2010) Analysing information flows and key mediators through temporal centrality metrics
- **Grindrod et al.** (2011) Communicability across evolving networks.