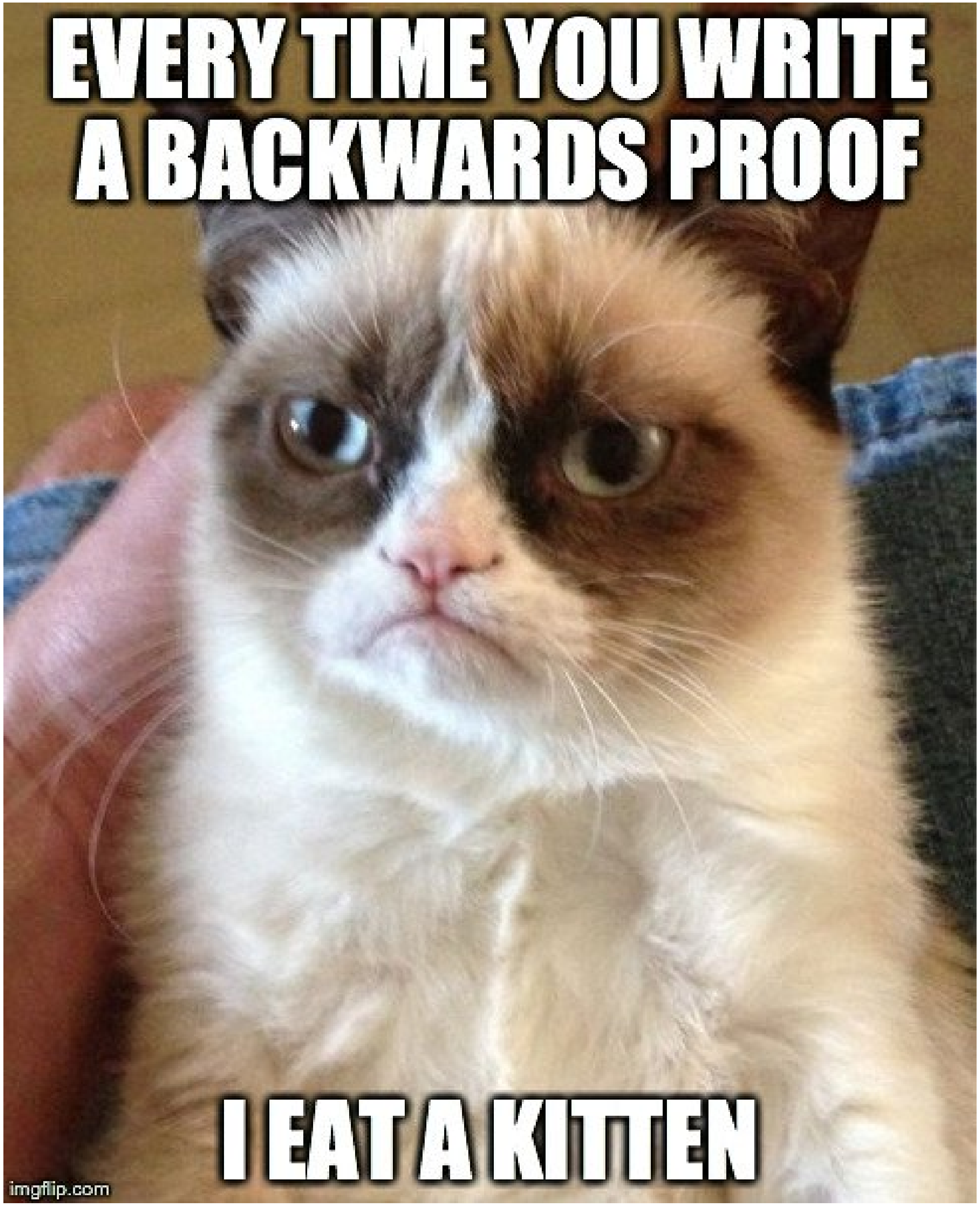
Graph Theory: Intro and Trees

CS 2800: Discrete Structures, Spring 2015

Sid Chaudhuri


EVERY TIME YOU WRITE
A BACKWARDS PROOF
I EAT A KITTEN
imgflip.com

This is ok

$$\sum_{i=0}^{n+1} F_i = \sum_{i=0}^{n} F_i + F_{n+1}$$
$$= F_{n+2} - 1 + F_{n+1}$$
$$= F_{n+1} + F_{n+2} - 1$$
$$= F_{n+3} - 1$$
$$= F_{(n+1)+2} - 1$$

(ok because equality is symmetric and transitive)

This is **NOT** ok

$$\sum_{i=0}^{n+1} F_i = F_{(n+1)+2} - 1$$

$$\Longrightarrow \quad \sum_{i=0}^{n} F_i + F_{n+1} = F_{(n+1)+2} - 1$$

$$\Longrightarrow \quad F_{n+2} - 1 + F_{n+1} = F_{(n+1)+2} - 1$$

$$\Longrightarrow \quad F_{n+1} + F_{n+2} - 1 = F_{(n+1)+2} - 1$$

$$\Longrightarrow \quad F_{n+3} - 1 = F_{(n+1)+2} - 1$$

$$\Longrightarrow \quad F_{(n+1)+2} - 1 = F_{(n+1)+2} - 1$$

... which is true, so QED ← No!

These $\Rightarrow$ symbols are implied if you omit them

Plea for the Day #1

Please read out your proofs in plain English and ask yourself if it makes sense

http://www.plainenglish.co.uk/

Plea for the Day #2

Plain English is often better than "mathy" notation

Instead of "$\forall d \in Days, Rainy(d) \Rightarrow Umbrella(d)$", say "If it's a rainy day, I will carry an umbrella"

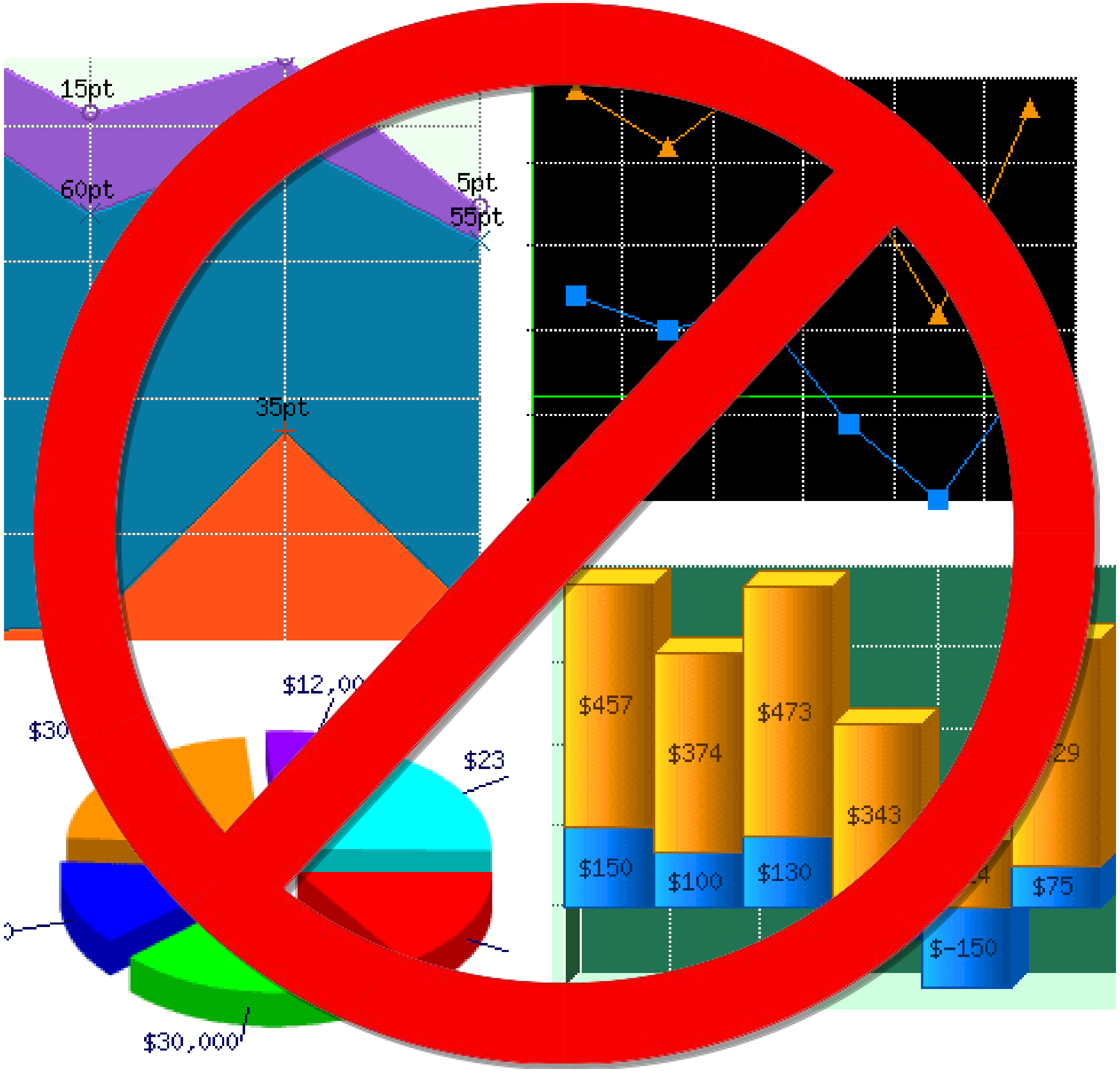
(Which is easier to read and debug?)

(But do be *precise* and *terse*)

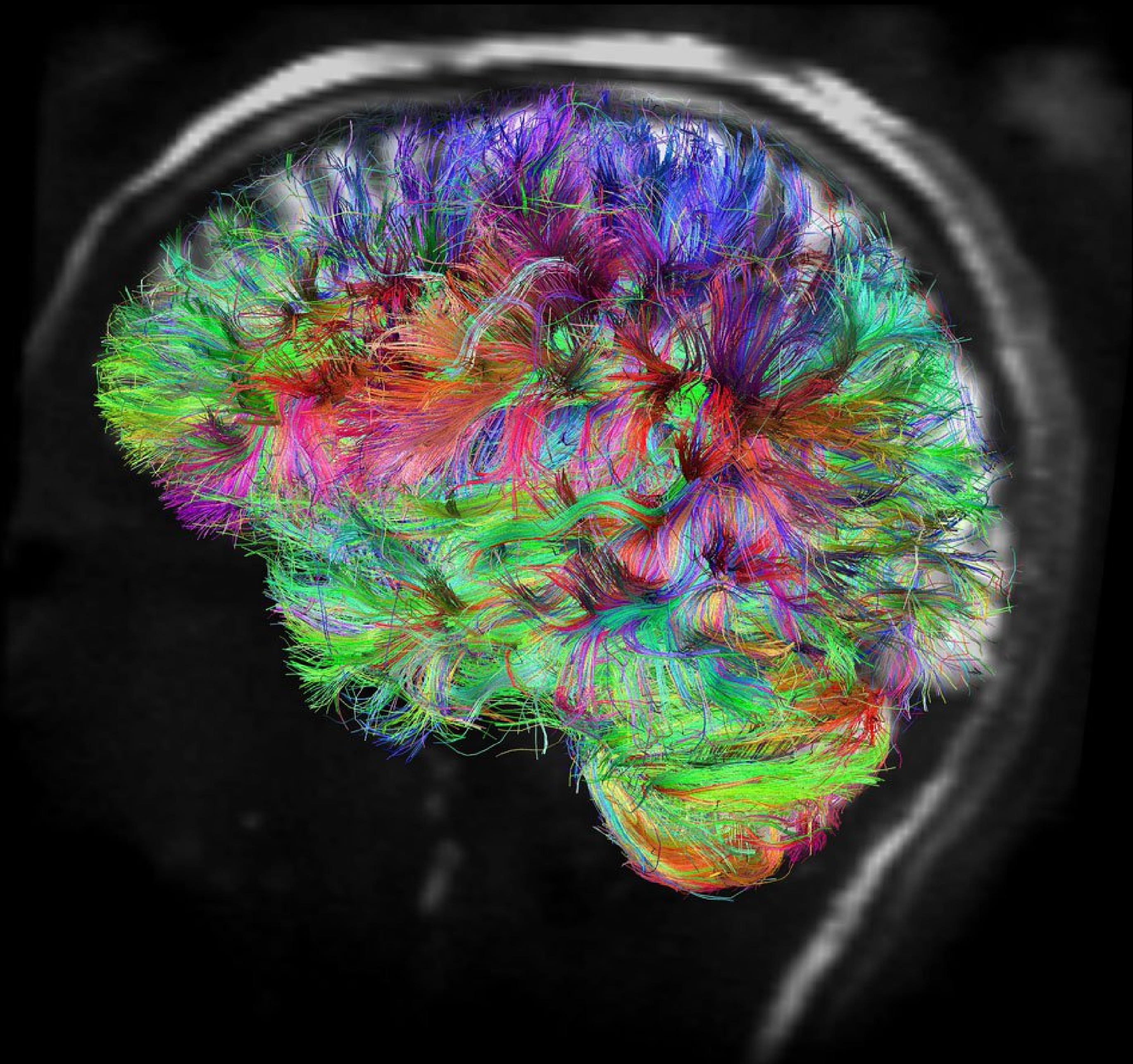
(This is an acquired skill – look at lots of well-written proofs)

<modest>… such as the homework and prelim solutions</modest>

These *aren't* the graphs we're interested in

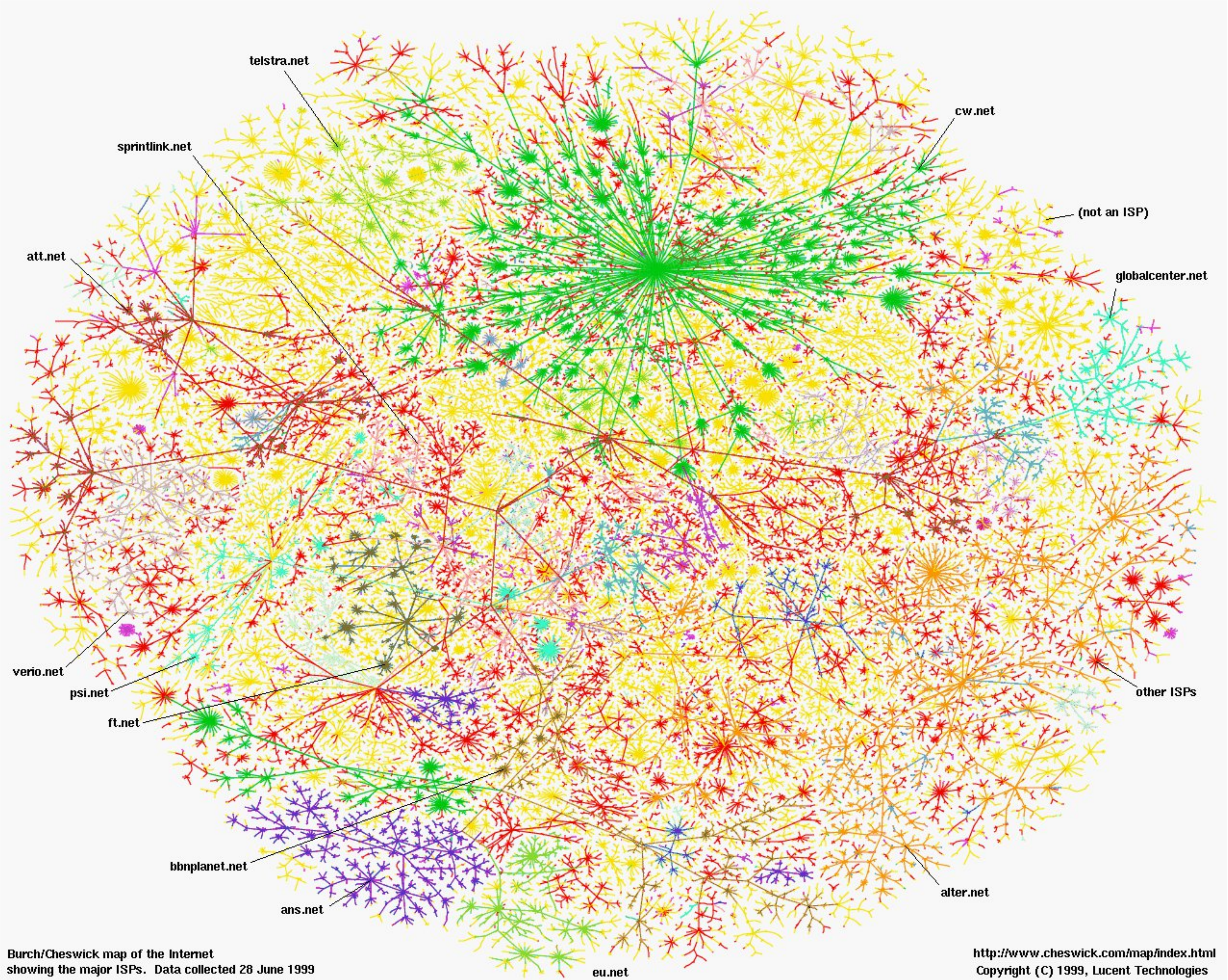


This is



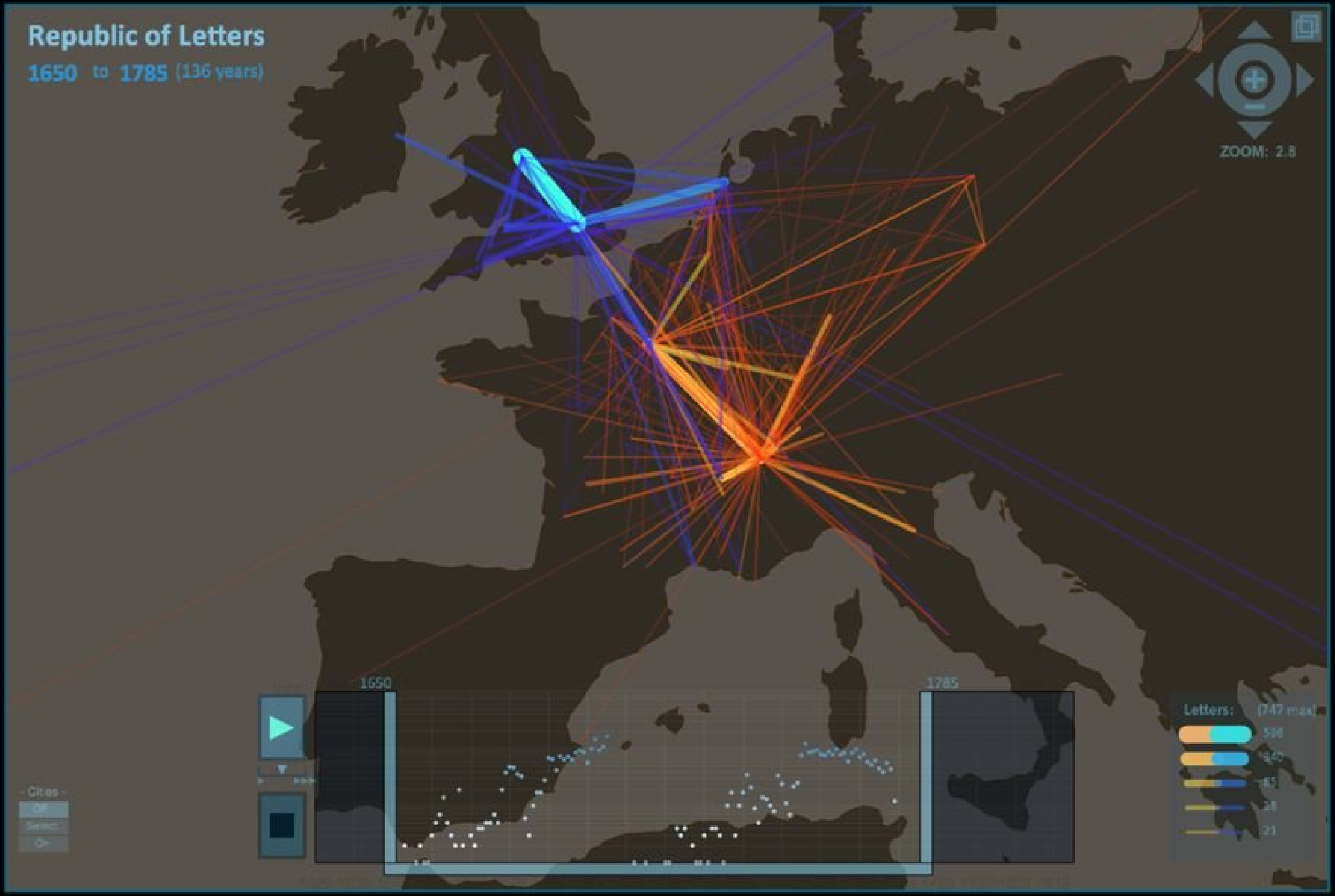
V.J. Wedeen and L.L. Wald, Martinos Center for Biomedical Imaging at MGH

And so is this



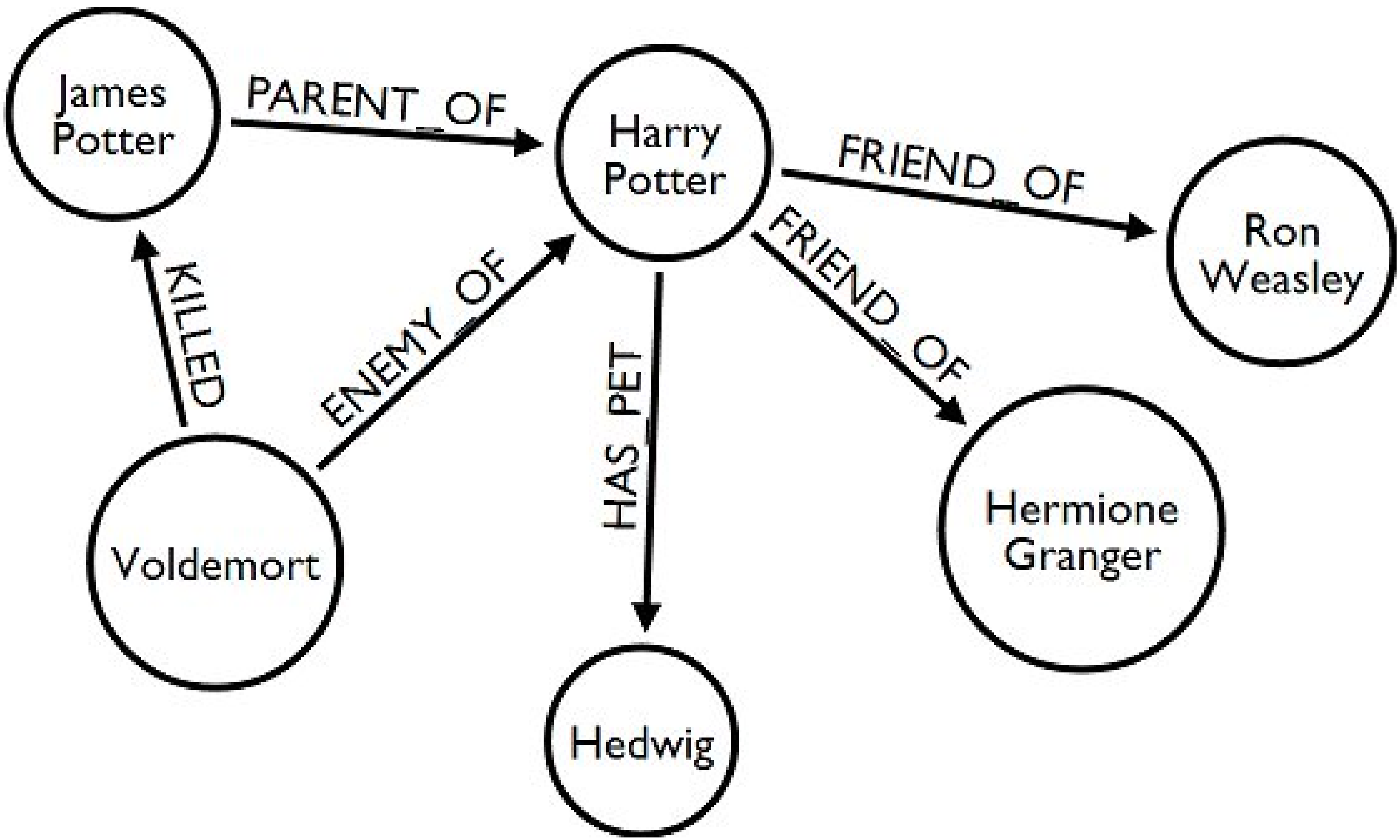
A social graph

An older social graph

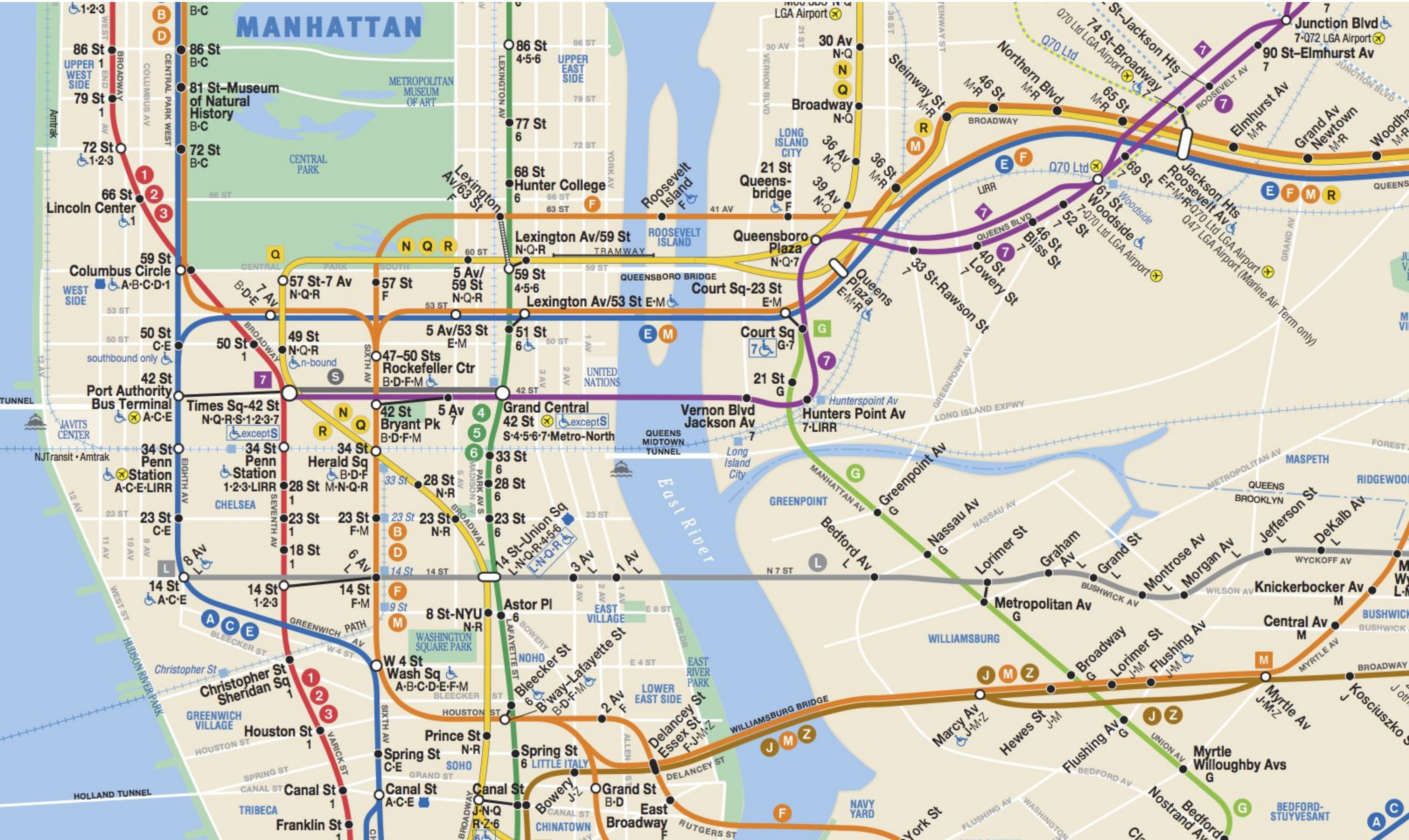


Locke's (blue) and Voltaire's (yellow) correspondence.
Only letters for which complete location information is available are shown.
Data courtesy the Electronic Enlightenment Project, University of Oxford.

A fictional social graph

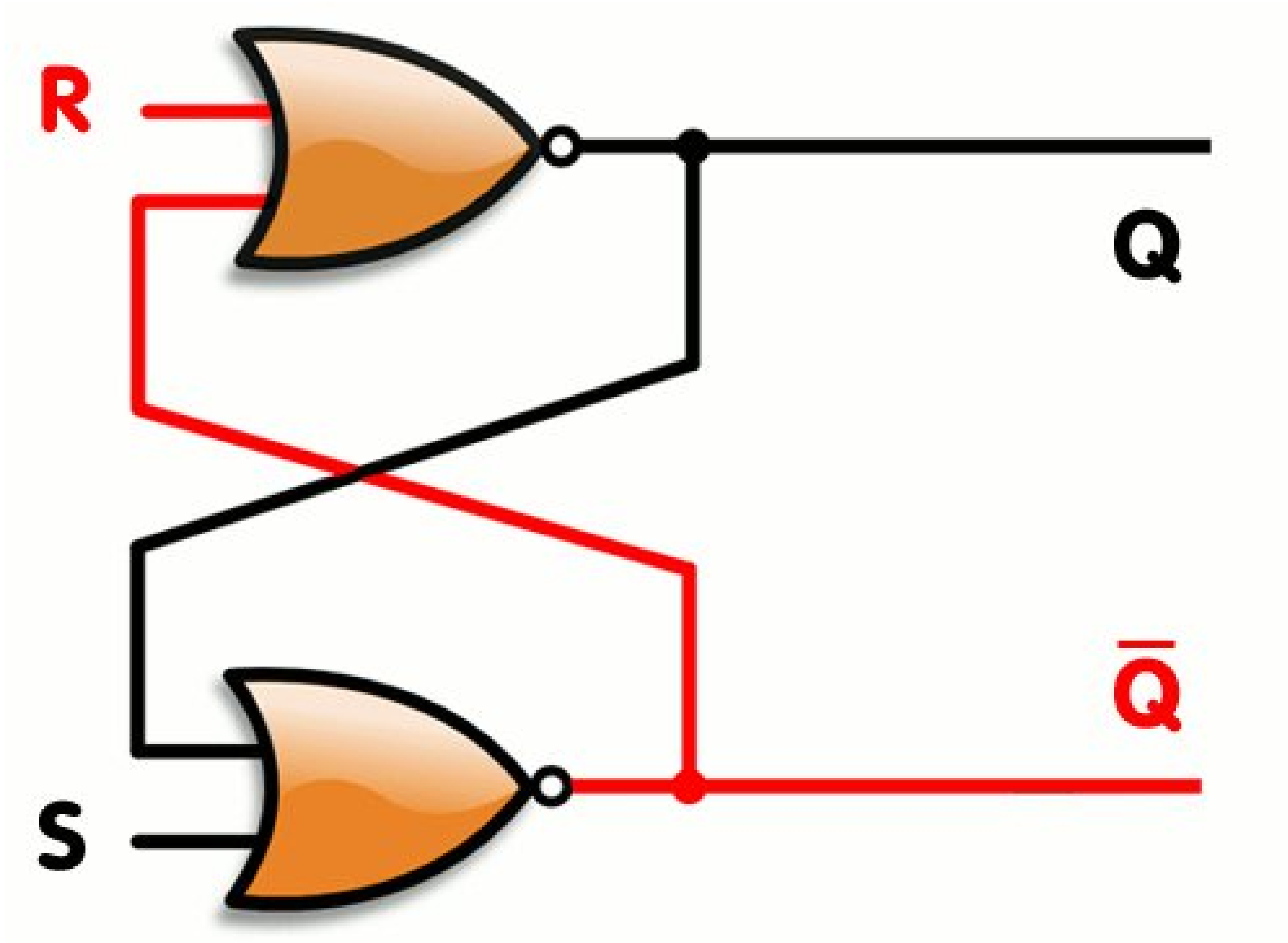


A transport graph

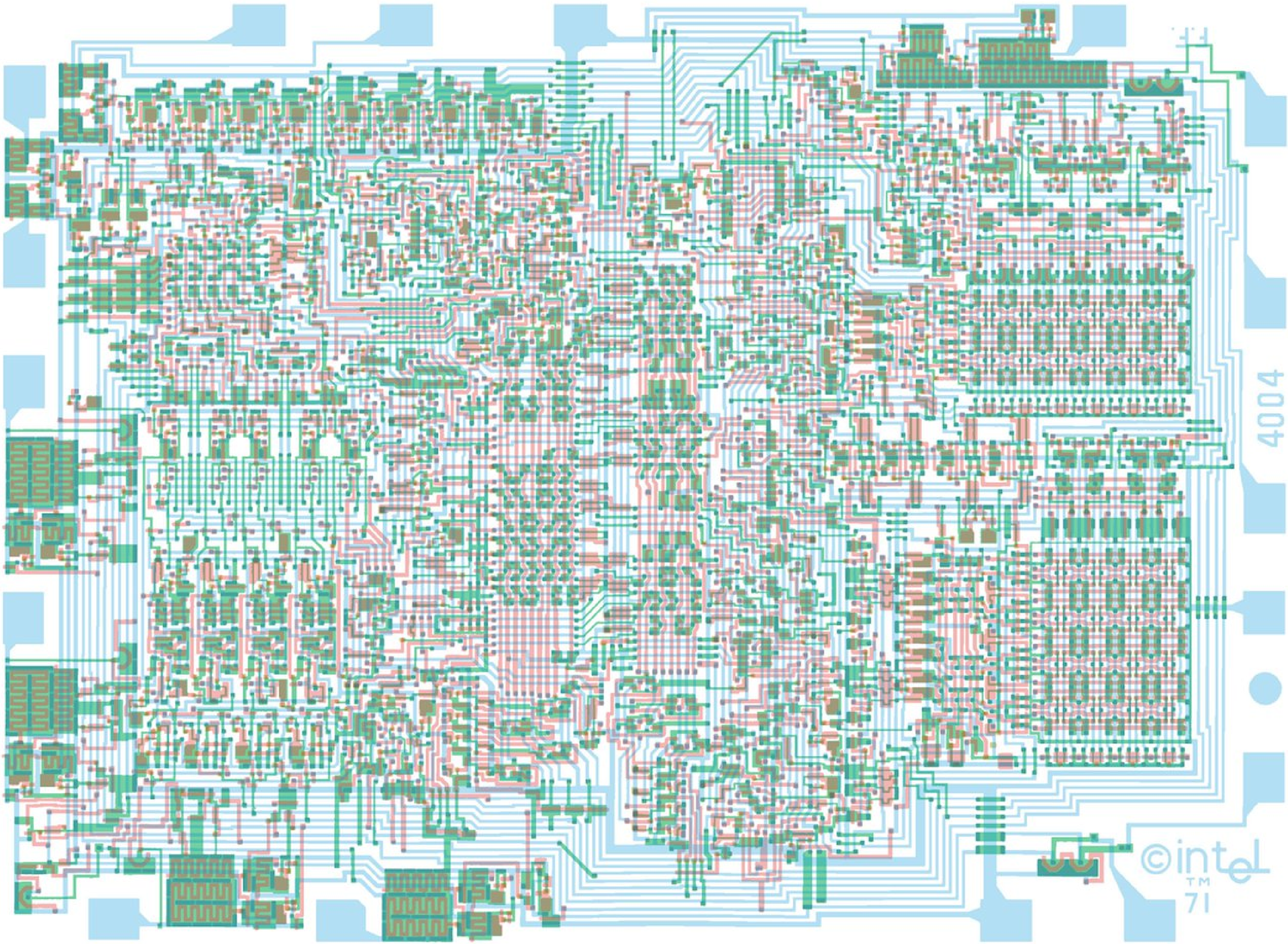


Another transport graph

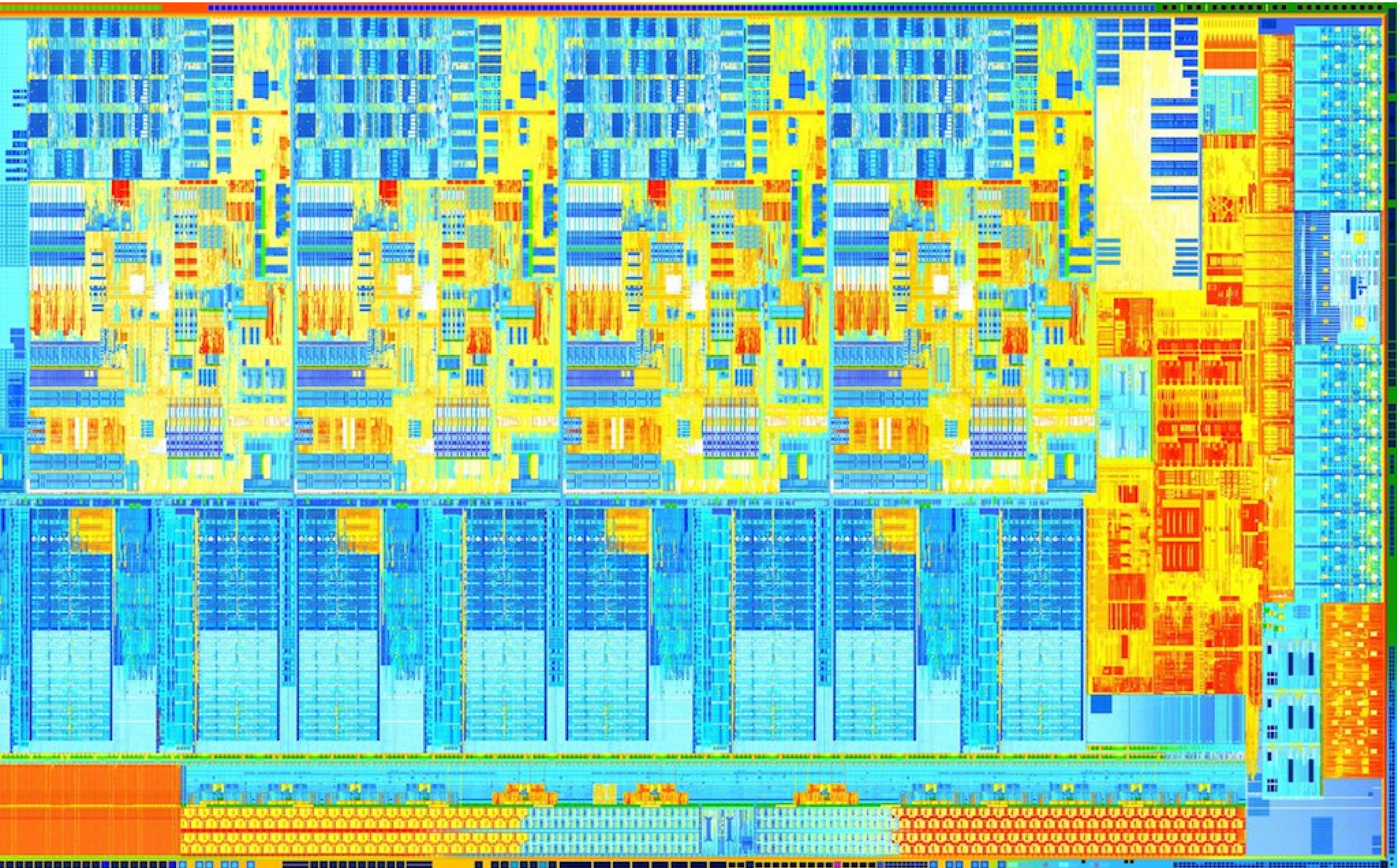
A circuit graph (flip-flop)



A circuit graph (Intel 4004)



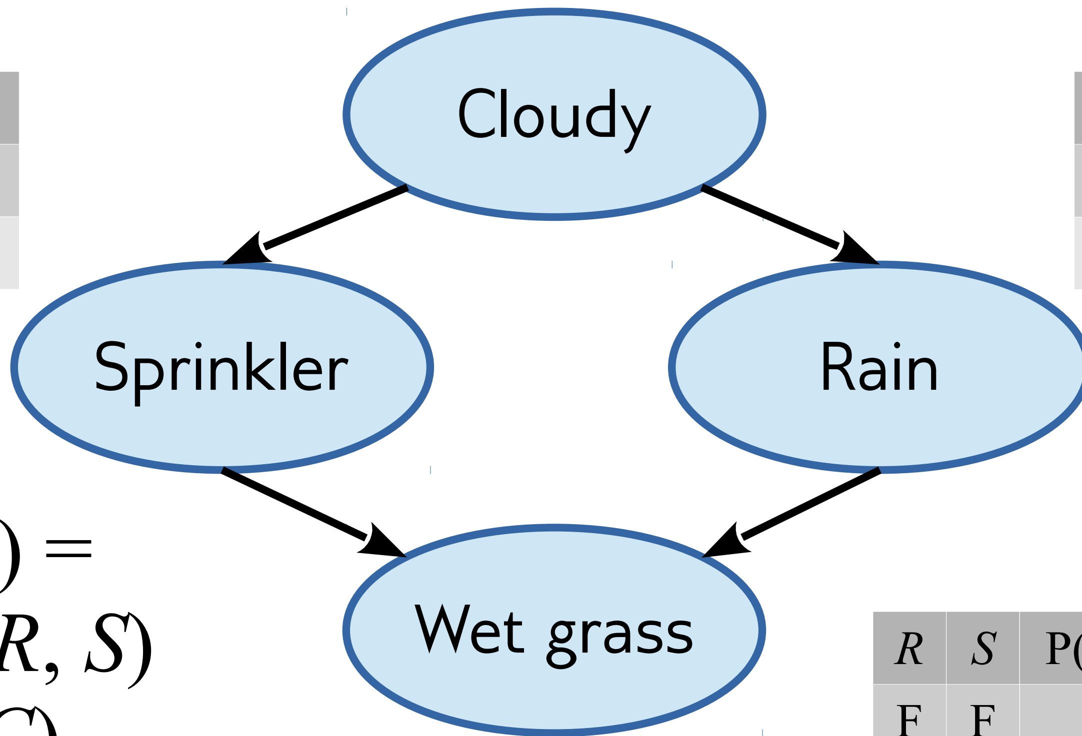
A circuit graph (Intel Haswell)



A probabilistic graphical model

P(*C*=F)	P(*C*=T)
0.5	0.5

C	P(*S*=F)	P(*S*=T)
F	0.5	0.5
T	0.9	0.1

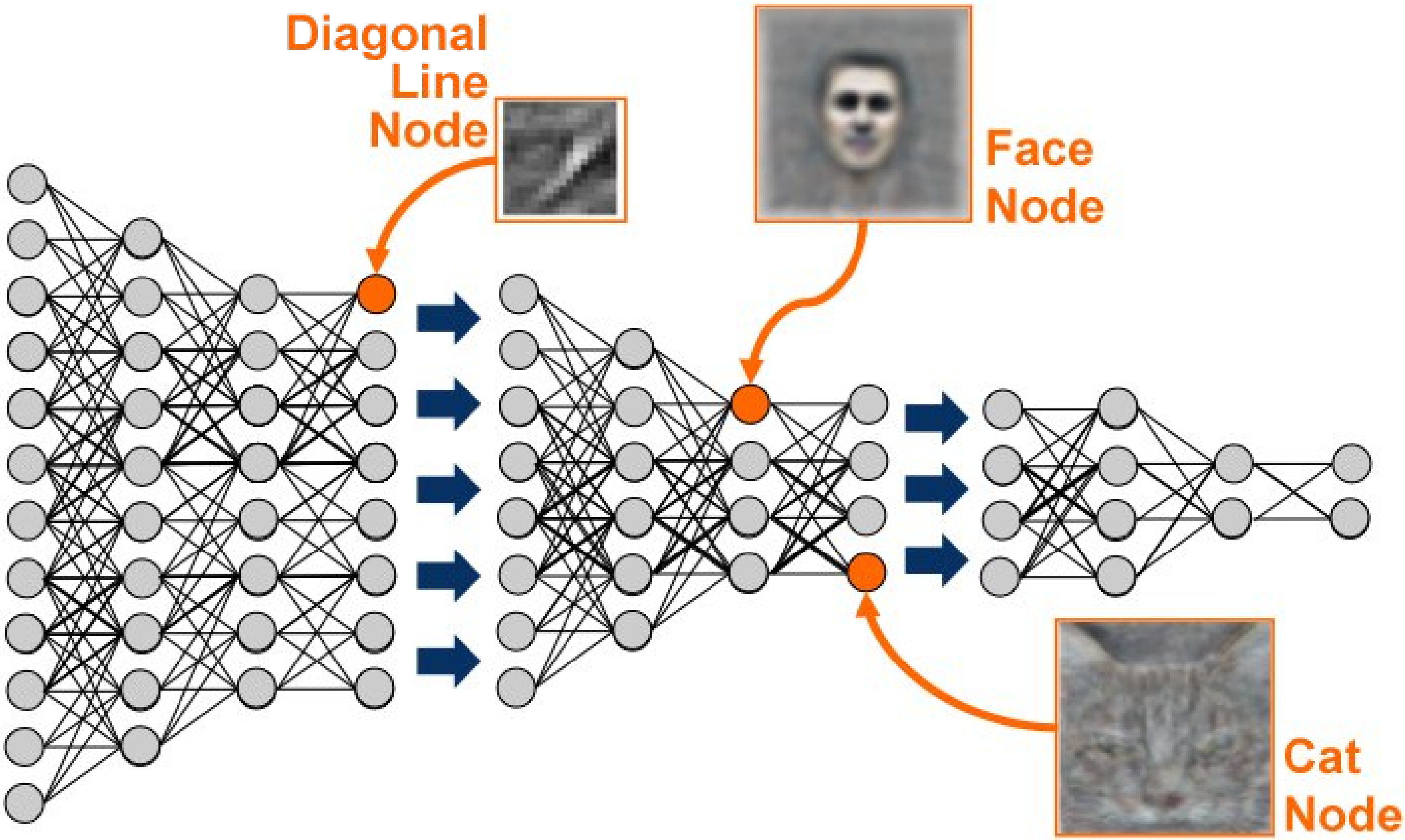


C	P(*R*=F)	P(*R*=T)
F	0.8	0.2
T	0.2	0.8

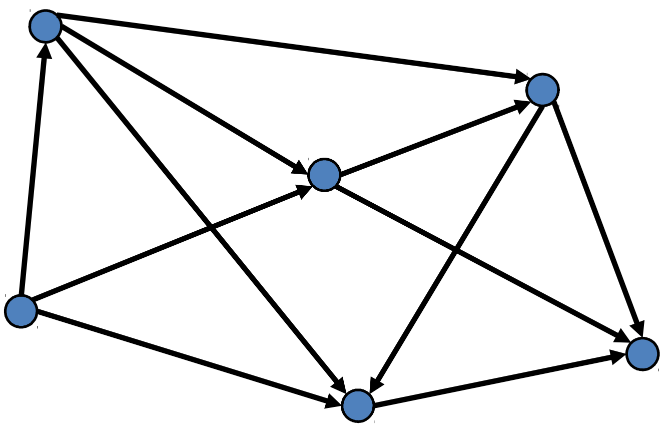
$$\begin{aligned} \mathrm{P}(W, R, S, C) = \\ & \mathrm{P}(W \mid R, S) \\ & \times \mathrm{P}(R \mid C) \\ & \times \mathrm{P}(S \mid C) \\ & \times \mathrm{P}(C) \end{aligned}$$

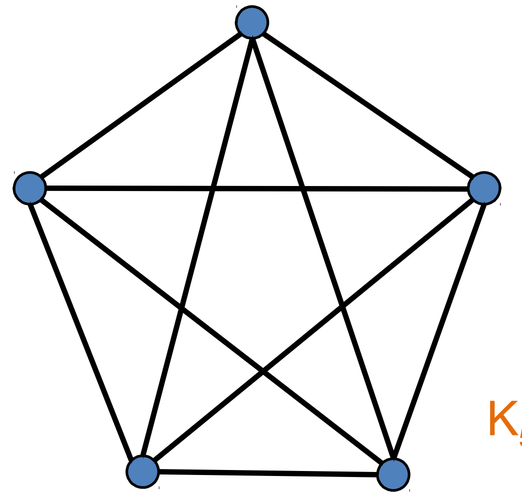
R	*S*	P(*W*=F)	P(*W*=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

This is a graph(ical model) that has learned to recognize cats

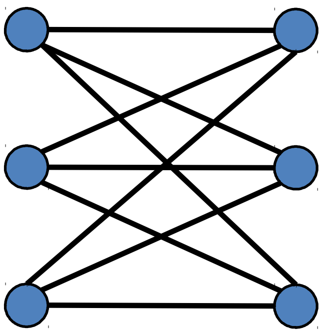


Some abstract graphs

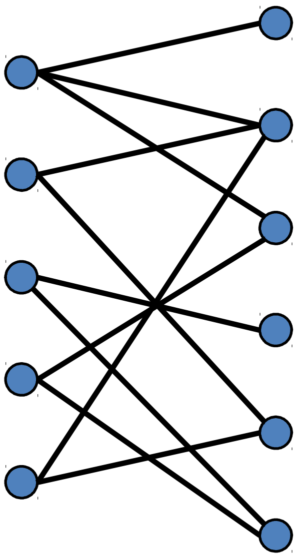


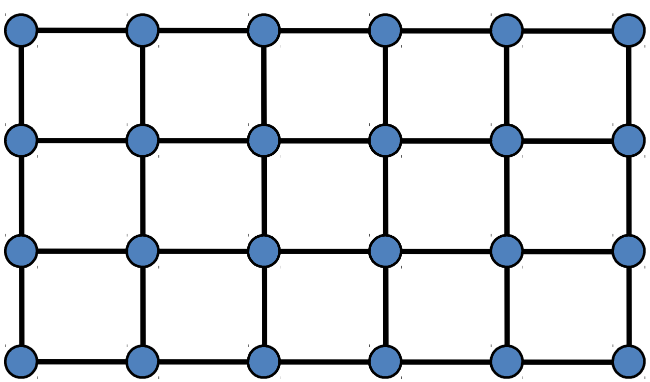


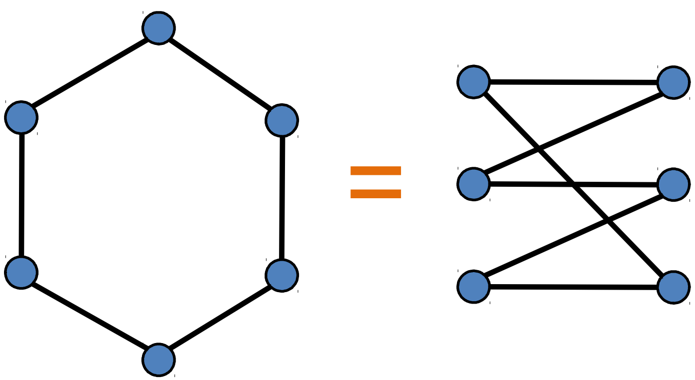
K_5



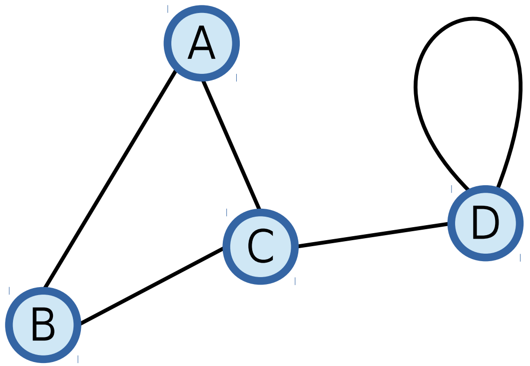
$K_{3,3}$



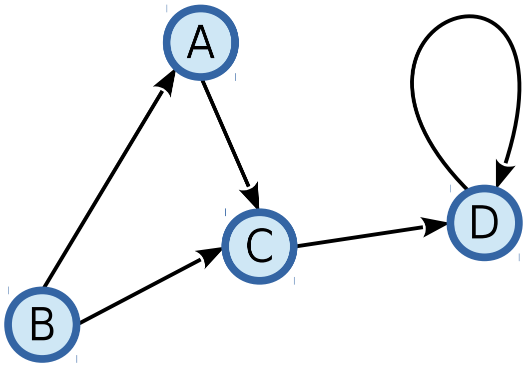




What is a graph?

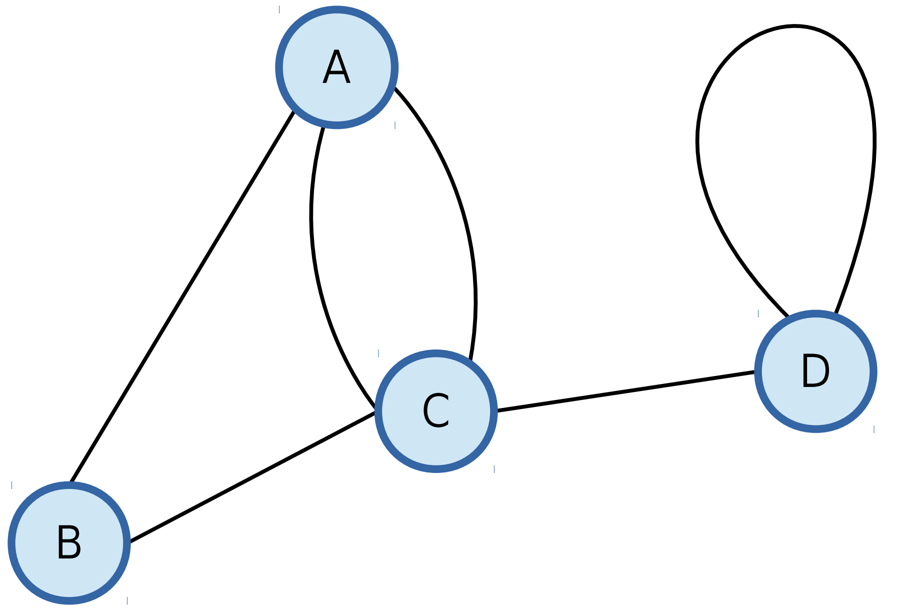


- An **undirected graph** $G = (V, E)$ consists of
  - A non-empty set of vertices/nodes V
  - A set of edges E, each edge being a set of one or two vertices (if one vertex, the edge is a self-loop)



- A **directed graph** $G = (V, E)$ consists of
  - A non-empty set of vertices/nodes V
  - A set of edges E, each edge being an *ordered* pair of vertices (the first vertex is the "start" of the edge, the second is the "end")
    - That is, $E \subseteq V \times V$, or E is a relation from V to V

Multigraphs



- Multiple edges between same pair of vertices
- Need a different representation (not a relation)

Adjacency, Incidence and Degree

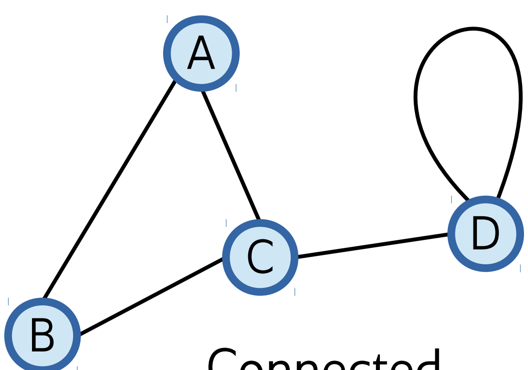
- Two vertices are adjacent iff there is an edge between them
- An edge is incident on both of its vertices
- *Undirected* graph:
  - Degree of a vertex is the number of edges incident on it
- *Directed* graph:
  - Outdegree of a vertex u is the number of edges leaving it, i.e. the number of edges (u, v)
  - Indegree of a vertex u is the number of edges entering it, i.e. the number of edges (v, u)

Paths

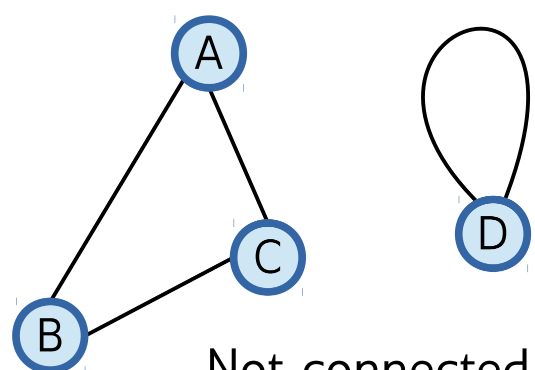
- A path is a sequence of vertices $v_0, v_1, v_2 \ldots v_n$, all different except possibly the first and the last, such that
  - (in an undirected graph) every pair $\{v_i, v_{i+1}\}$ is an edge
  - (in a directed graph) every pair (v_i, v_{i+1}) is an edge
- Alternatively, a path may be defined as a sequence of distinct edges $e_0, e_1, e_2 \ldots e_n$ such that
  - Every pair e_i, e_{i+1} shares a vertex
  - These vertices are distinct, except possibly the first & the last
  - If the graph is directed, then the end vertex of e_i is the start vertex of e_{i+1} (the "arrows" point in a consistent direction)
- We will use these interchangeably
- A cycle is a path with the same first and last vertex

Connectedness

- An *undirected* graph is connected iff for every pair of vertices, there is a path containing them
- A *directed* graph is strongly connected iff it satisfies the above condition for all ordered pairs of vertices (for every u, v, there are paths from u to v and v to u)
- A *directed* graph is weakly connected iff replacing all directed edges with undirected ones makes it connected



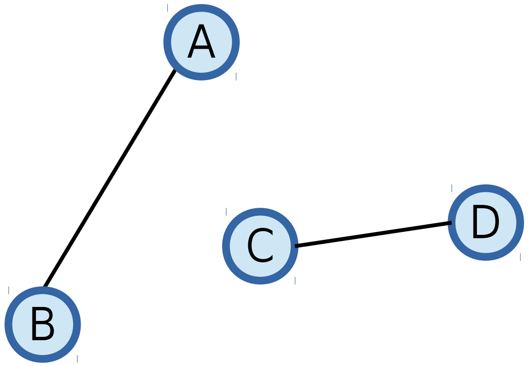
Connected

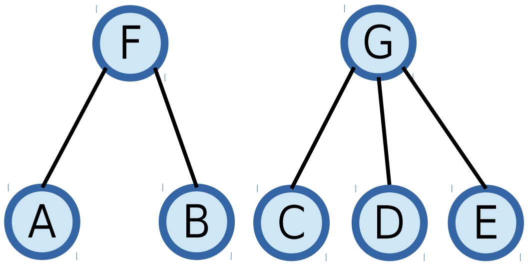


Not connected

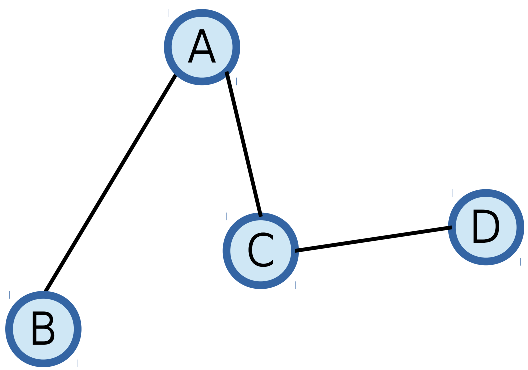
Trees

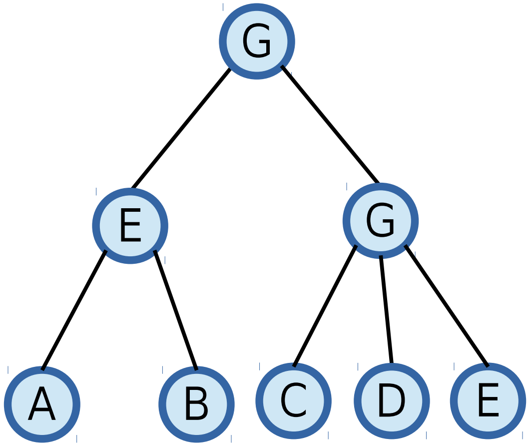
- A forest is an undirected graph with no cycles





- A tree is a connected forest (← definition)





Identifying trees

- An undirected graph G on a finite set of vertices is a tree iff **any two** of the following conditions hold
  - $|E| = |V| - 1$
  - G is connected
  - G has no cycles
- Any two of these imply the third
- There are many other elegant characterizations of trees

Proof Sketches

- If G is connected and lacks cycles, then $|E| = |V| - 1$
  - Show that such a graph always has a vertex of degree 1
  - Use induction, repeatedly removing such a vertex
- If G is connected and $|E| = |V| - 1$, then it lacks cycles
  - Show that a connected graph has a spanning tree
  - Apply the $|E| = |V| - 1$ formula to the spanning tree
- If G lacks cycles and $|E| = |V| - 1$, then it is connected
  - If disconnected, must have ≥ 2 connected components, each of which must be a tree
  - Sum vertex and edge counts over connected components, show that they don't add up

Cayley's Formula

- For any positive integer n, the number of trees on n labeled vertices is n^{n-2}