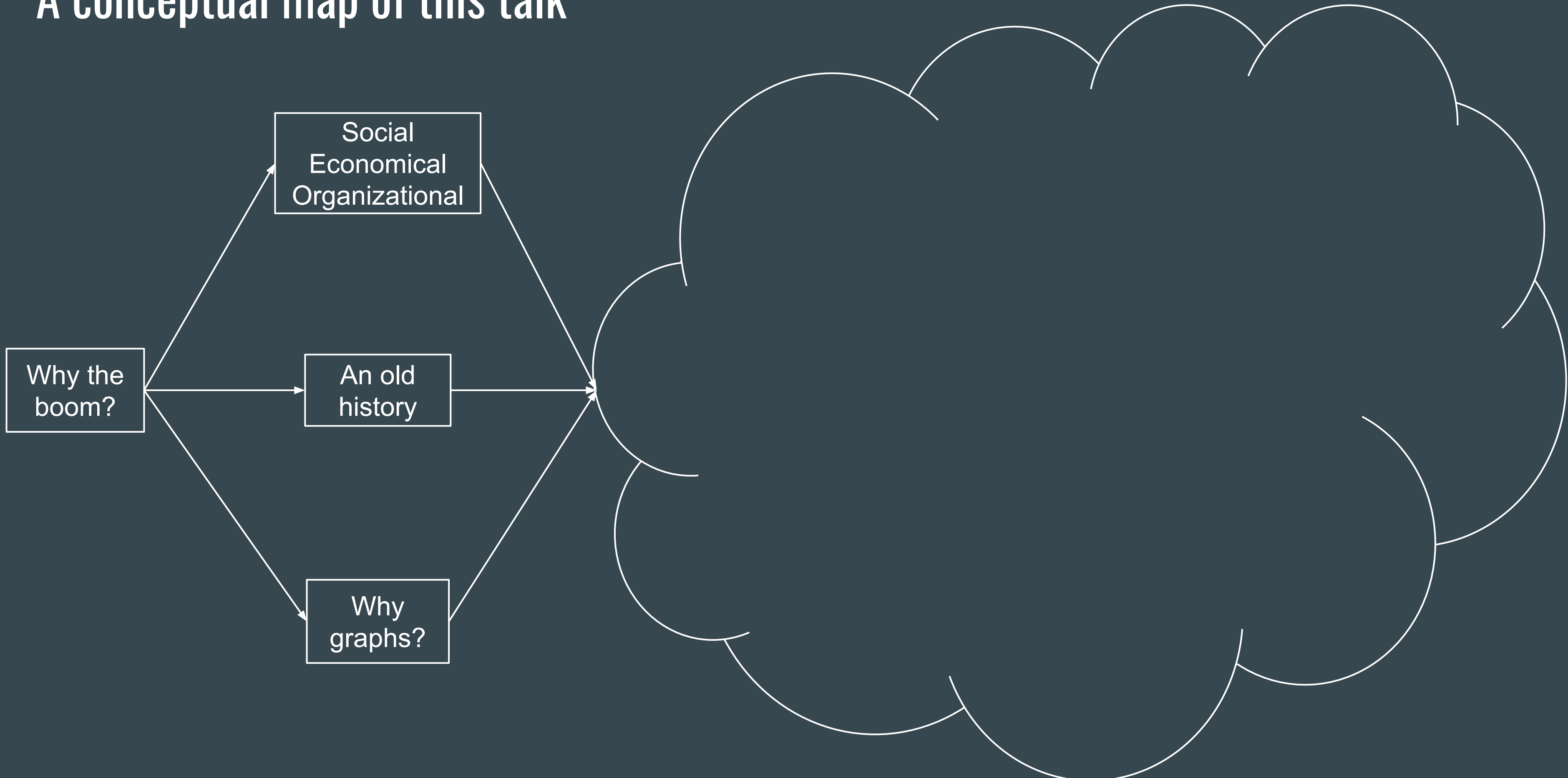
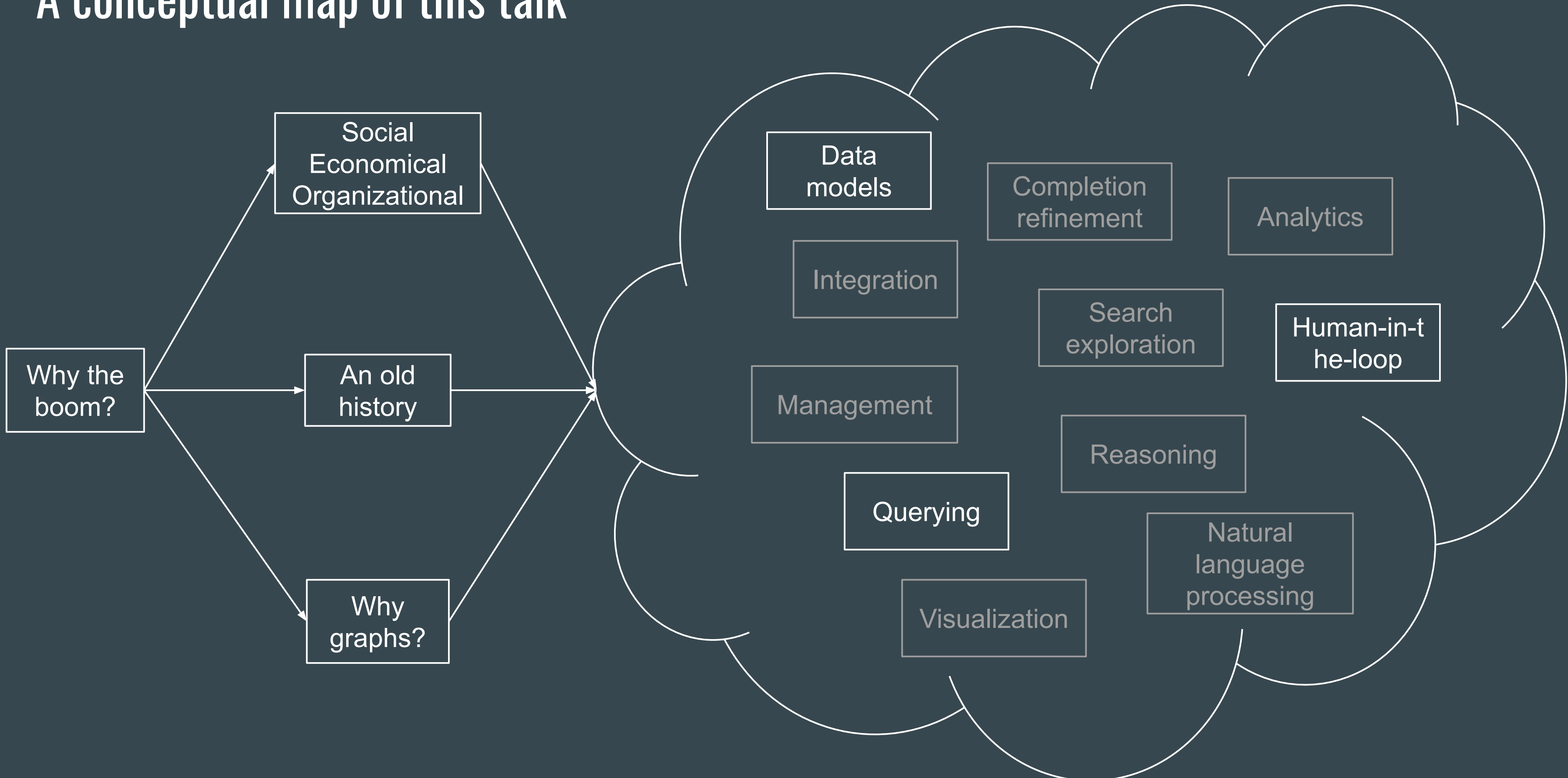
Querying in the Age of Graph Databases and Knowledge Graphs

Marcelo Arenas, Claudio Gutierrez and Juan Sequeda

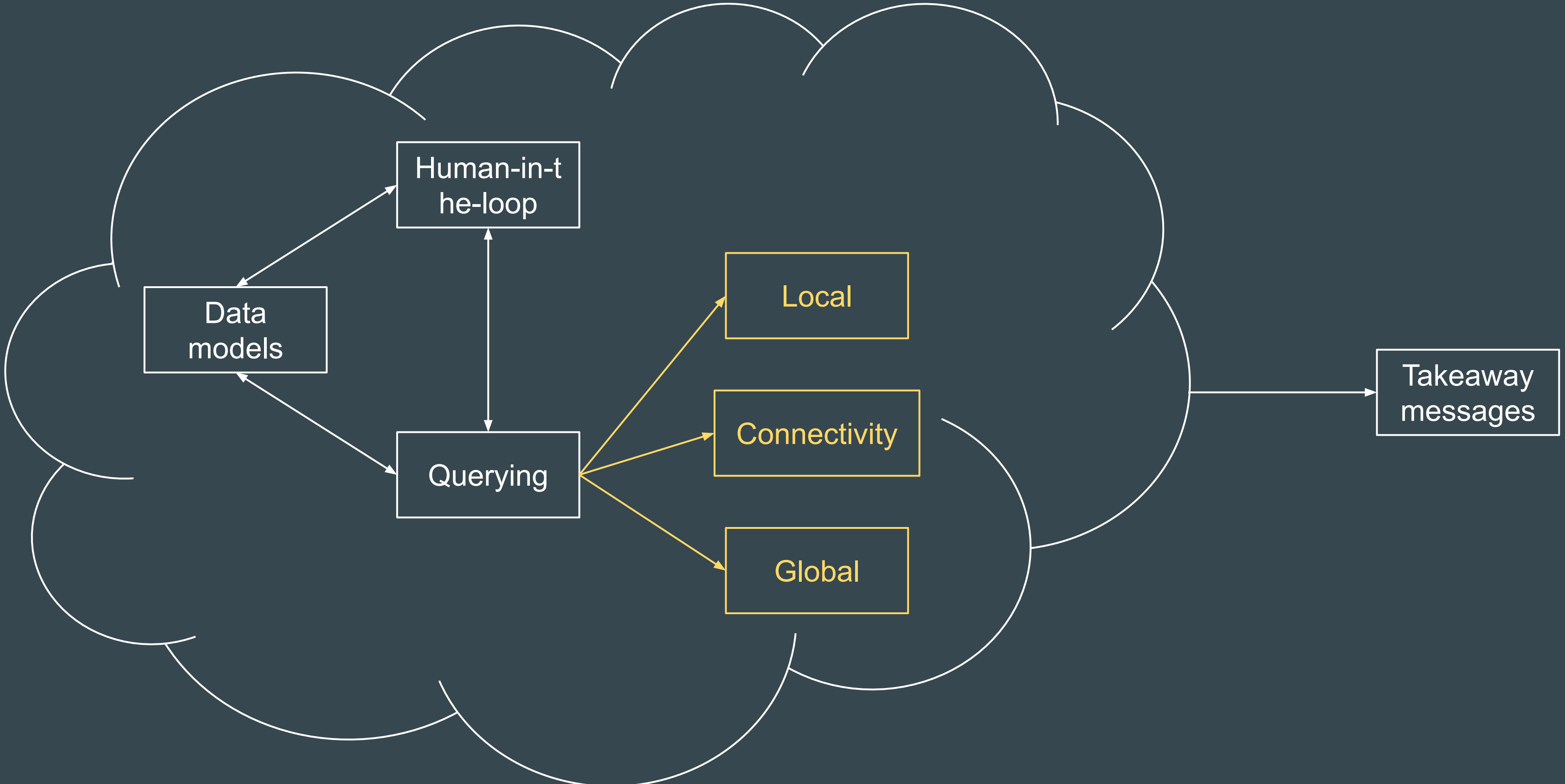
A conceptual map of this talk



A conceptual map of this talk



A conceptual map of this talk



A necessary digression:
Why are we here?
Why graphs everywhere?
…

Real World: Big Tech Giants

SEARCH

Introducing the Knowledge Graph: things, not strings

Amit Singhal
SVP, Engineering

Published May 16, 2012

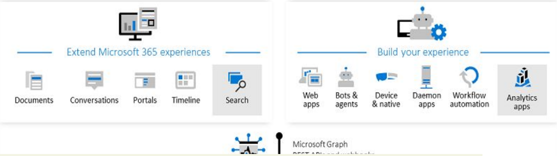
Search is a lot about discovery—the basic human need to learn and broaden your horizons. But searching still requires a lot of hard work by you, the user. So today I'm really excited to launch the Knowledge Graph, which will help you discover new information quickly and easily.

Apple is shoring up Siri for its next generation of intelligent devices

John Mannes @johnmannes / 3:43 pm CDT • May 16, 2017

Comment

Microsoft 365 Platform



Product Graph

product graph

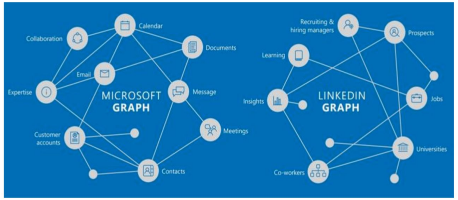
❑Mission: To answer any question about products and related knowledge in the world

	Data model	Size of the graph
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million

Building The LinkedIn Knowledge Graph

Qi He October 6, 2016

Share Tweet Share



Authors: Qi He, Bee-Chung Chen, Deepak Agarwal

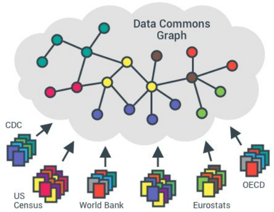
Real World: Large KG

WIKIDATA

NSF's Open Knowledge Network

Data Commons

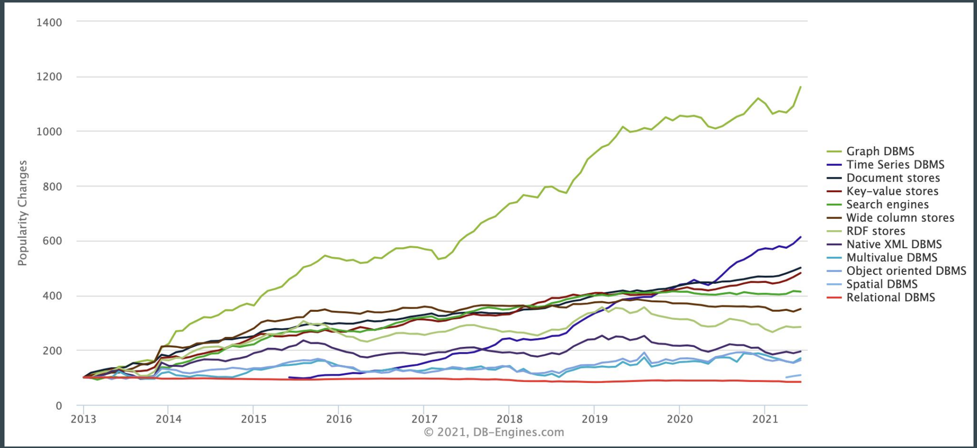
Data Commons is an open knowledge repository that combines data from public datasets using mapped common entities. It includes tools to easily explore and analyze data across different datasets without data cleaning or joining.



DBpedia

yago
select knowledge

Real World: Market Growth



https://db-engines.com/en/ranking_categories

Market Guide for Graph Database Management Solutions

Published 24 May 2021

By 2025, graph technologies will be used in 80% of data and analytics innovations, up from 10% in 2021, facilitating rapid decision making across the enterprise.

https://www.gartner.com/doc/4001808

Trend No. 8: Graph relates everything

Graph forms the foundation of modern data and analytics with capabilities to enhance and improve user collaboration, machine learning models and explainable AI. Although graph technologies are not new to data and analytics, there has been a shift in the thinking around them as organizations identify an increasing number of use cases. In fact, as many as 50% of Gartner client inquiries around the topic of AI involve a discussion around the use of graph technology.

Gartner Top 10 Data and Analytics Trends for 2021

https://www.gartner.com/smarterwithgartner/gartner-top-10-data-and-analytics-trends-for-2021/

Real World: Not just the Big Tech Giants

BBC

Most content management done with aid of knowledge graph.
Coordinates journalist's work. Powers article recommendations

EA

"Intelligent Content Ecosystem" (videos, games, articles, ...)
Meaningful product and article recommendations, age ratings across multiple jurisdictions and languages, along with justifications

UnitedHealth Group

Member profiles from single system, near real-time, including contact timeline and care-path recommendations.
Claimed $150 million per year savings

Source: Mohammed Aaser, CDO of McKinsey. Presentation at Knowledge Graph Conference May 2021
https://knowledgegraphconference.vhx.tv/videos/mohammed-aaser-future-of-enterprise-data-management

Real World: Data Product Companies too

WUNDERMAN THOMPSON CHOREOGRAPH

Supports $2bn marketing agency for all customer insights
Integration of 100s of millions of datapoints from 1000s of sources
Development efficiency: “do in 12 hours what we couldn’t in 6 months”

reonomy

“Unprecedented Products”: ownership portfolio of commercial real-estate
Development efficiency: new dataset live in 5 days with junior dev

DIFFBOT

Multiple data products based on unstructured web-base data: news, organizations, people

enigma

Gathers data from many sources to create a private Small & Medium Business data product

Source: Mohammed Aaser, CDO of McKinsey. Presentation at Knowledge Graph Conference May 2021
https://knowledgegraphconference.vhx.tv/videos/mohammed-aaser-future-of-enterprise-data-management

A bit of history about knowledge and data

[Gutierrez and Sequeda 2021]

Knowledge

Graphical Representation of Knowledge

Automation of Reasoning

Searching in Spaces

Information Retrieval of Unstructured Sources

Resolution Principle

General Solving Problem

Dijkstra Shortest Path

A* Search

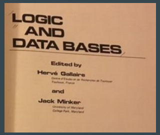
Semantic Networks

Frames

Conceptual Graphs

Logic Programming

Prolog



Workshop on Logic and Data Bases, 1977

Description Logic

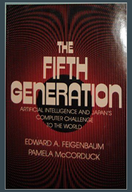
AI Winter

KL-ONE/ LOOM/CLASSIC

KADS

MCC Austin Cyc

Japanese 5th Generation Project



Expert Systems

DL Reasoners

OWL

DAML+OIL

RDF

SPARQL

Schema.org

Wikidata

"Knowledge Graph" 2012

Linked Data

DBpedia

Big Data

MapReduce 2004

BigTable 2006

NoSQL

Dynamo 2007

Datalog

Deductive Databases

OLAP

Data Integration

XML

XQUERY

OEM

GraphDB

Languages and Systems for Data

COBOL

Network Databases

ER

Relational Algebra

System R Ingres

QUEL SQL

Web

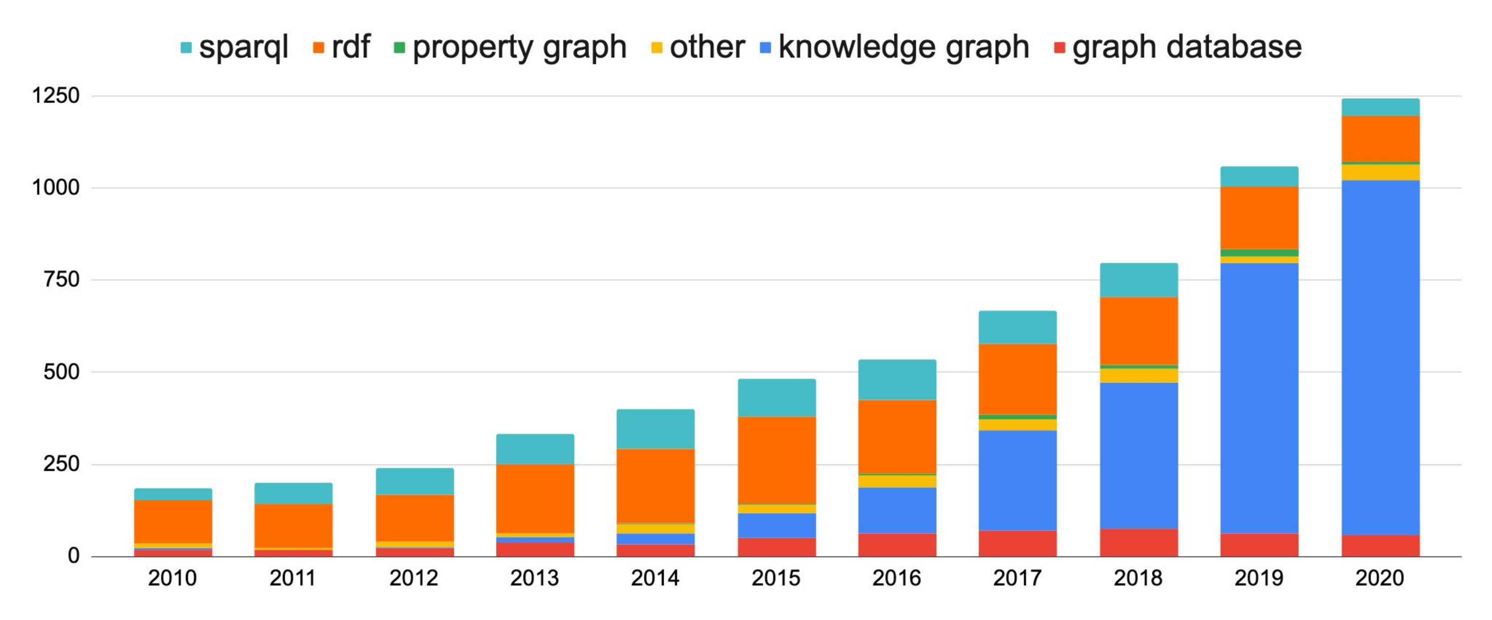
Graph Golden Era

OODB

Data

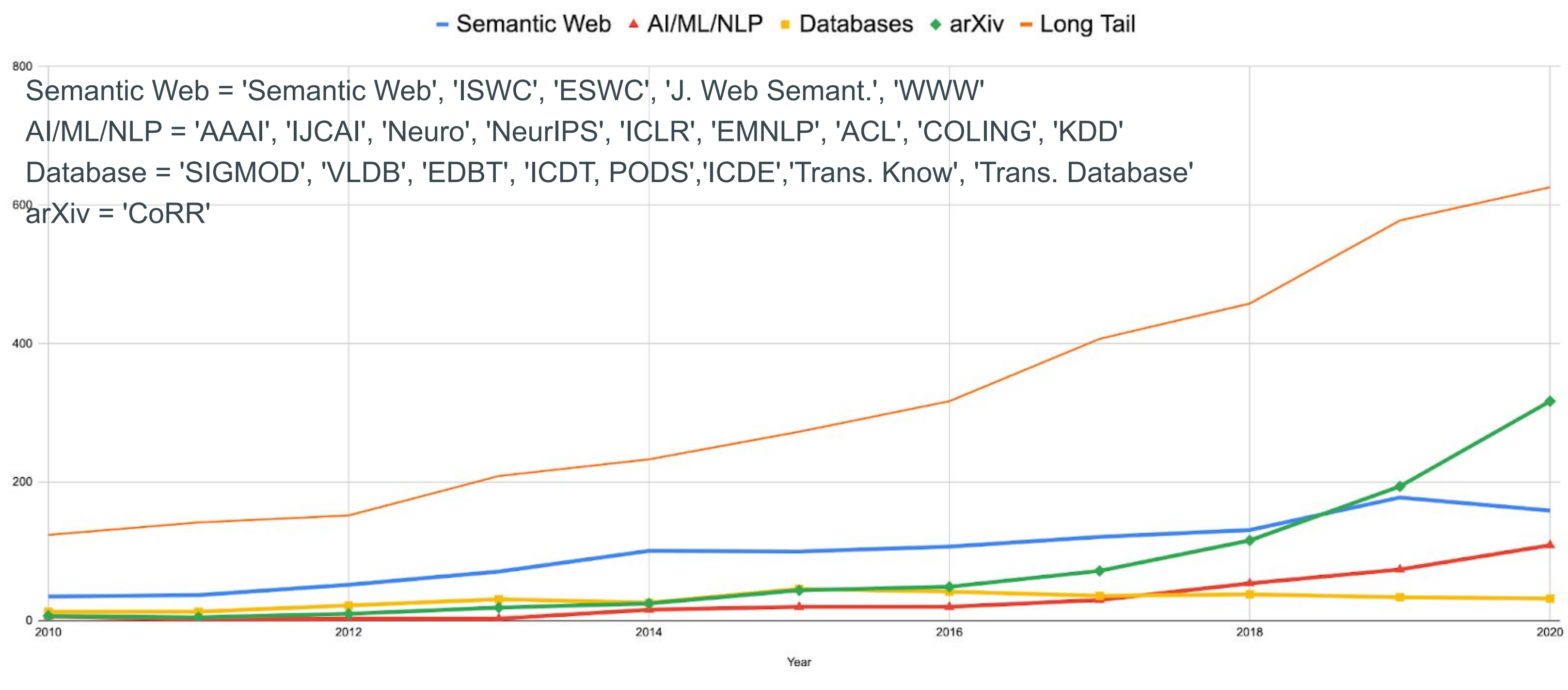
1950 | 1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

KG types of papers (per DBLP)



Number of papers where sparql/rdf/knowledge graph/graph database/property graph appears in the title

Where are KG papers being published (per DBLP)



Summary of graph data today, past and future

- Real World
  - Knowledge graphs are not just for the Tech Giants
  - KG and Graph Databases are already in many places and it will keep growing
- Rich History
  - This isn’t new, it’s been boiling up for a while
- Academic Interest
  - Steady academic interest

Why did graphs become so relevant for data and knowledge?

…

What is new today?

1. Graphs were long ago recognized as prime representation media for **knowledge**

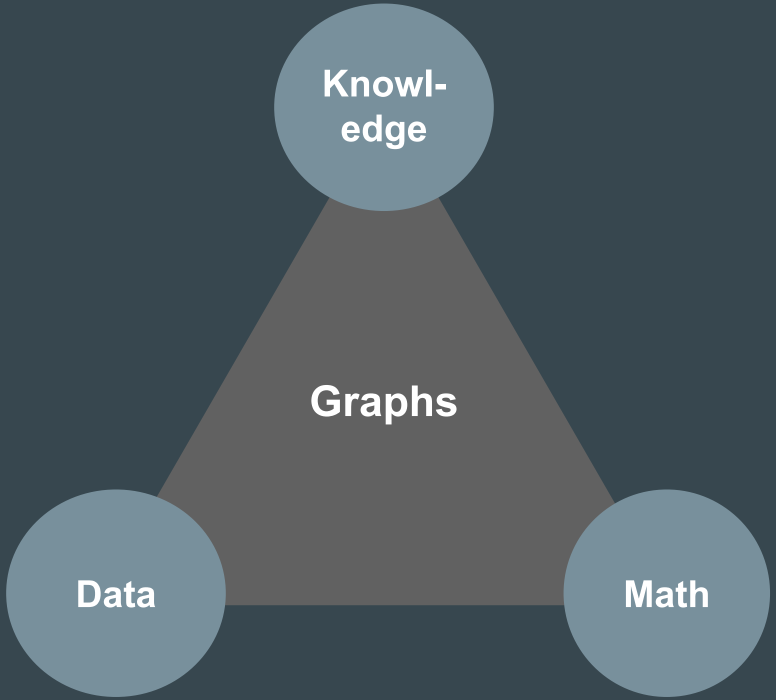
2. The network-like intrinsic characteristic of **data** was also well known

3. Graphs are well known and studied mathematical objects

The novelty today is the integration of these three previously disjoint trends

An outline of this part

- Graphs and knowledge
- Graphs and data
- Graphs as mathematical structures

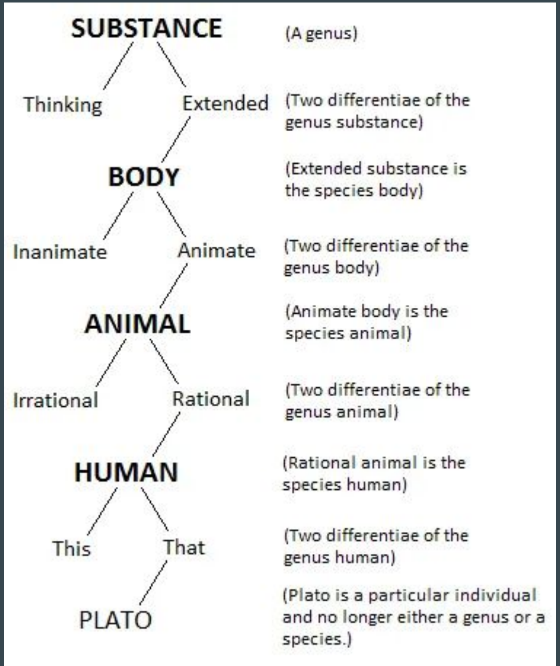


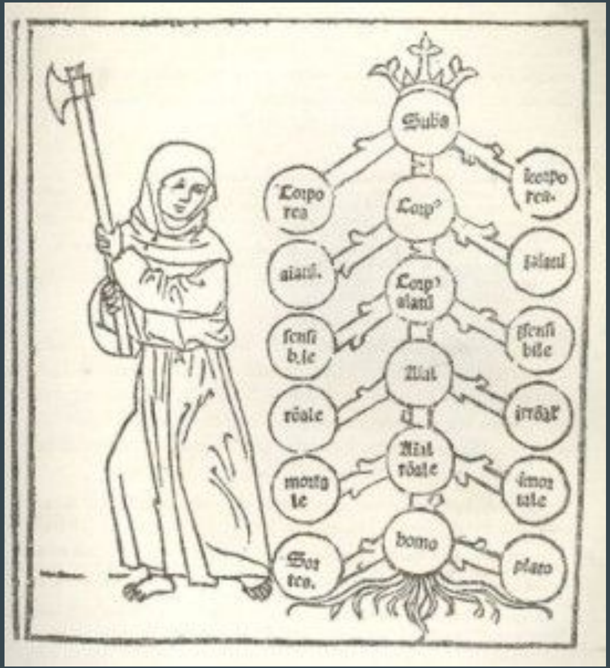
Knowledge and graphs: an old history

- Aristotle and categories
- Lull and tree of knowledge
- Routes in maps
- Chemical graphs
- Semantic Networks

  ...
- Graph databases
- Knowledge graphs

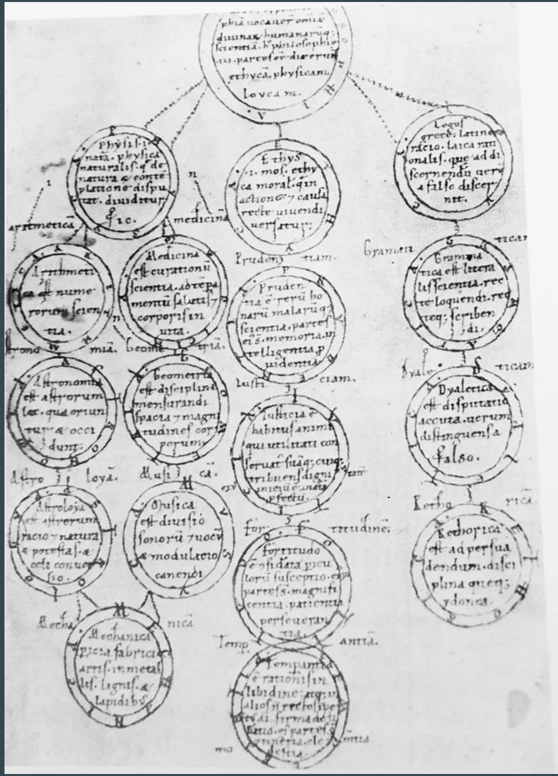
Tree-shaped visualization of Aristotle categories





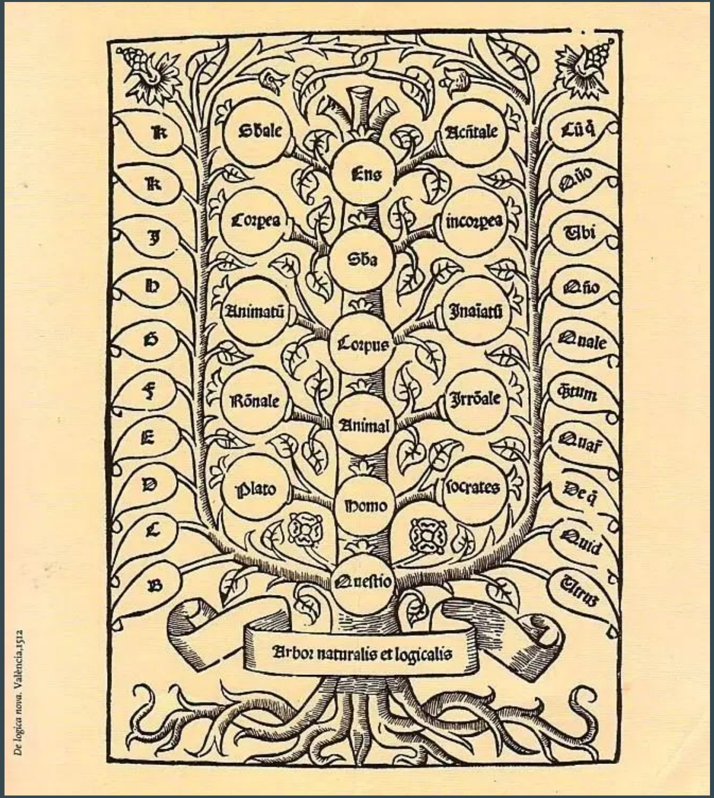
Porphyrian Tree (left, 4th century) and its “deletion” on the left (16th century).
(This and following illustrations taken from Scott B. Weingart: https://scottbot.net/knowledge/)

Labeled nodes, labeled edges and graphs (no only trees)



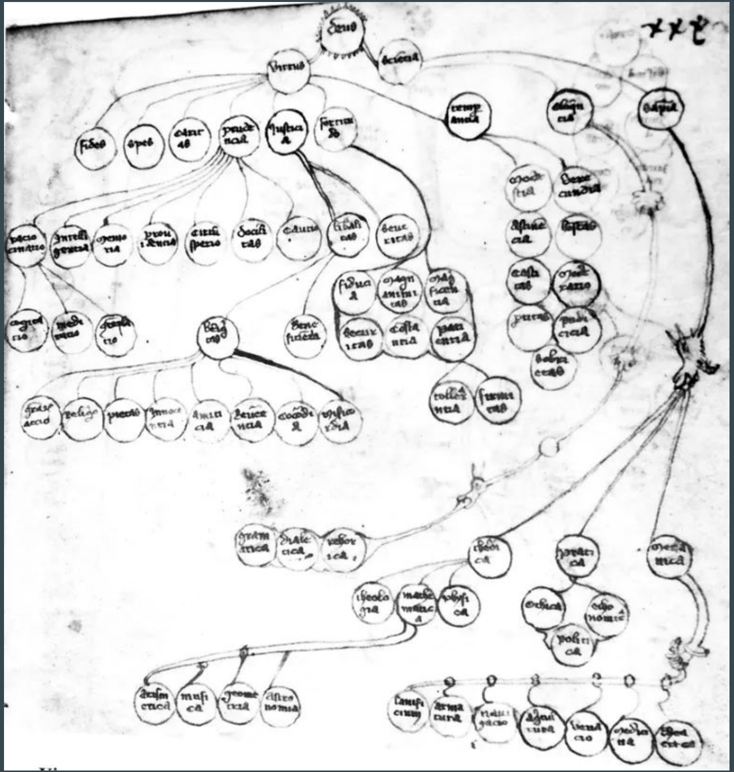
A twelfth century manuscript splitting philosophy into dichotomies (ibidem)

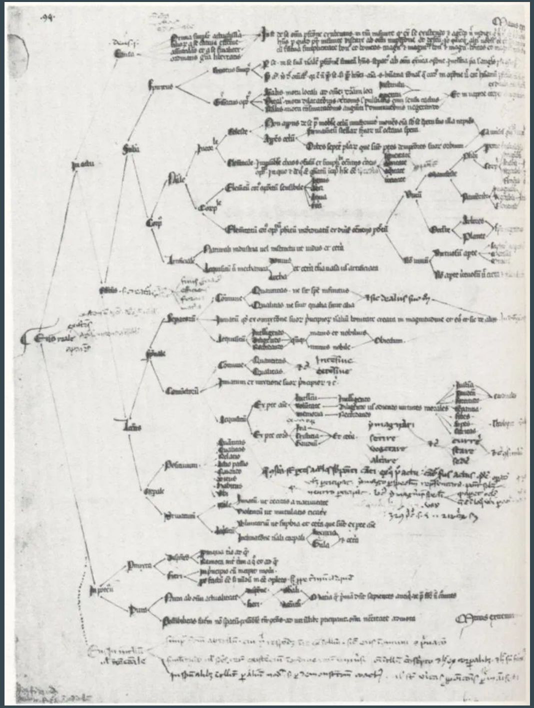
An ordered tree



Tree of Knowledge (Ramon Llull)

Problems with the material form of the representation

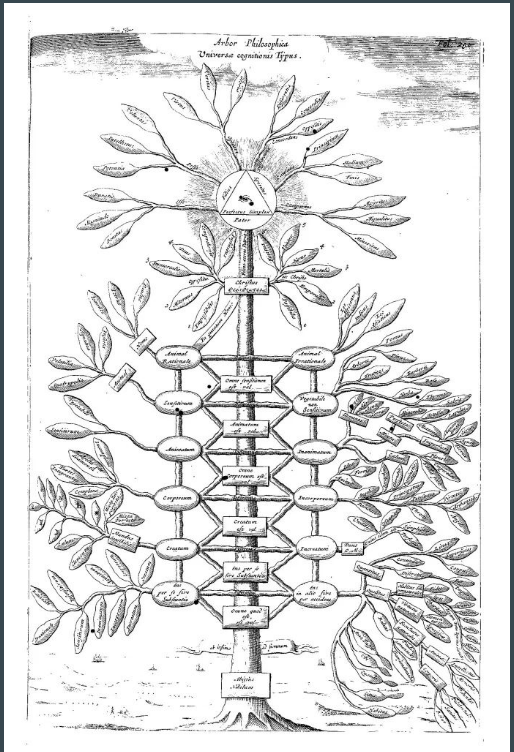




14th century diagrams

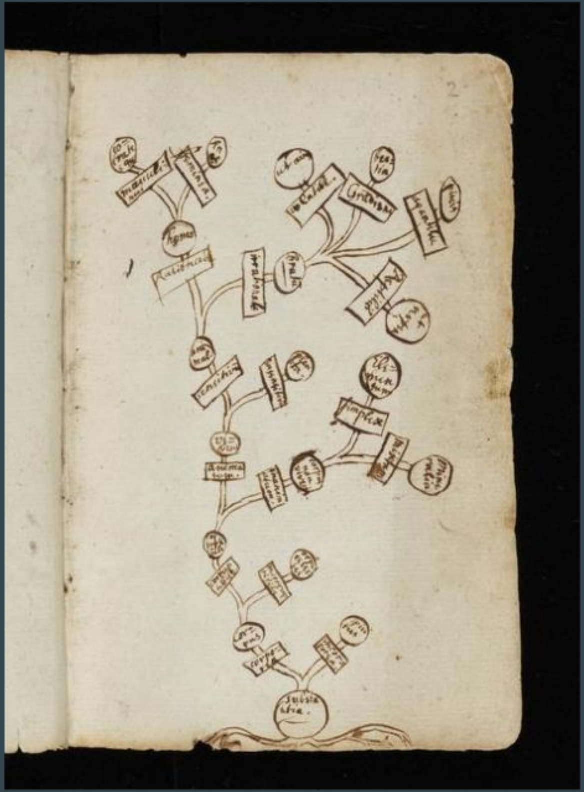
Non-digital XML: Italy and the Andes (circa 15th century)

Representation of different types of nodes, edges and complex graph structure



Athanasius Kircher’s Philosophical tree representing all branches of knowledge (1669)

Diagrams in Newton’s notebook

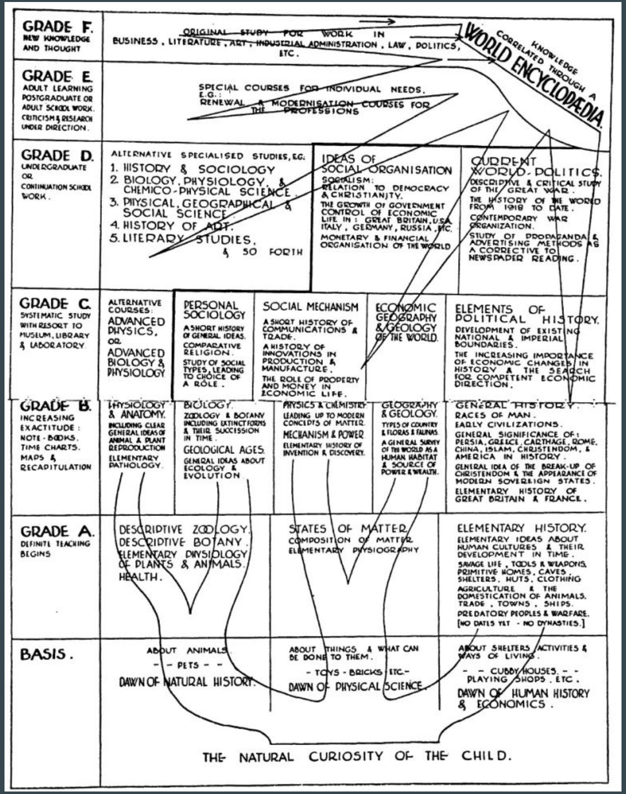


Newton’s ‘Trinity College Notebook’ (MS Add. 3996)

It was used by him as an undergraduate, from about 1661 to 1665.

http://cudl.lib.cam.ac.uk/view/MS-ADD-03996/5

1938's conceptual graph



H.G. Wells describing how students ought to learn in 1938.

Tree-like polymer topologies

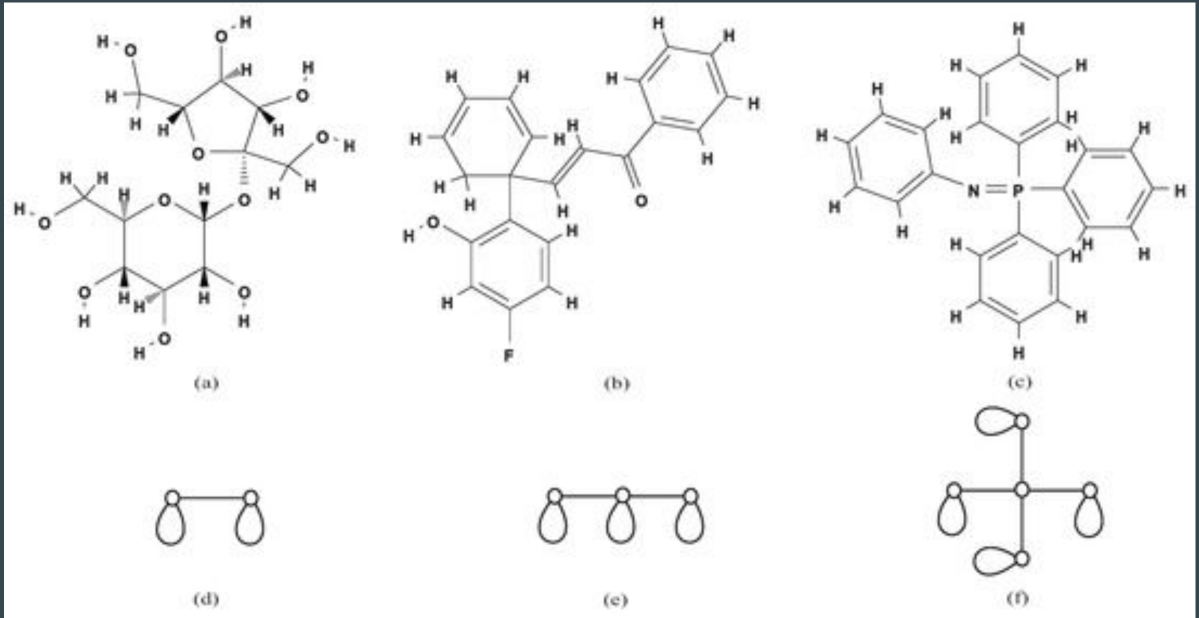
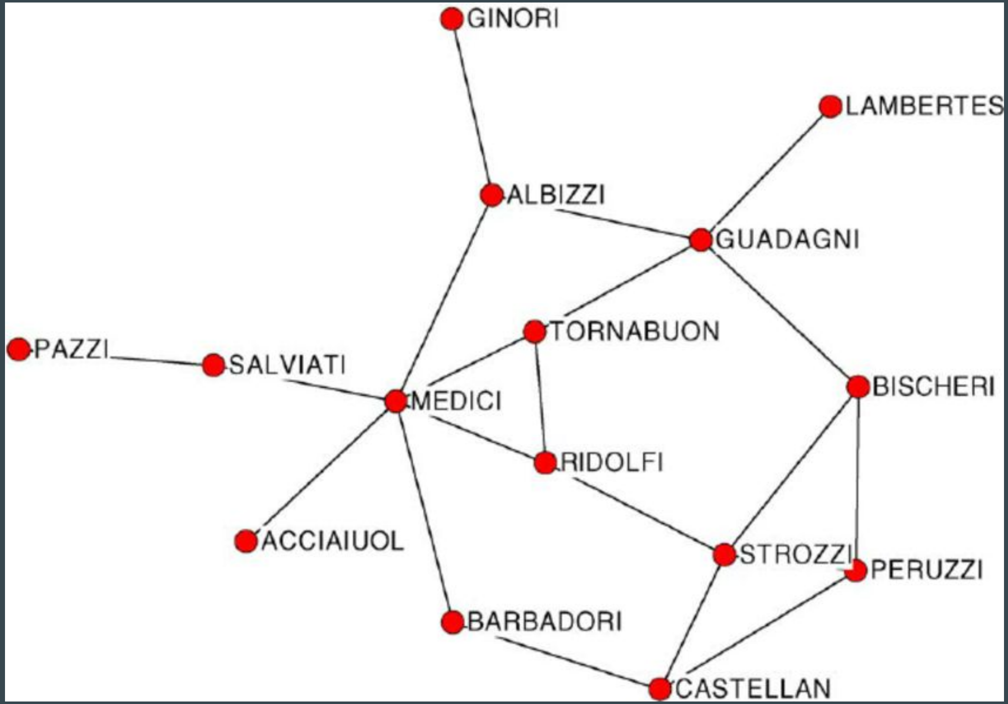


Image in: Azam, N.A.; Shurbevski, A.; Nagamochi, H. Enumerating Tree-Like Graphs and Polymer Topologies with a Given Cycle Rank. *Entropy* **2020**

Social / semantic / linguistic networks



Florentine families' network

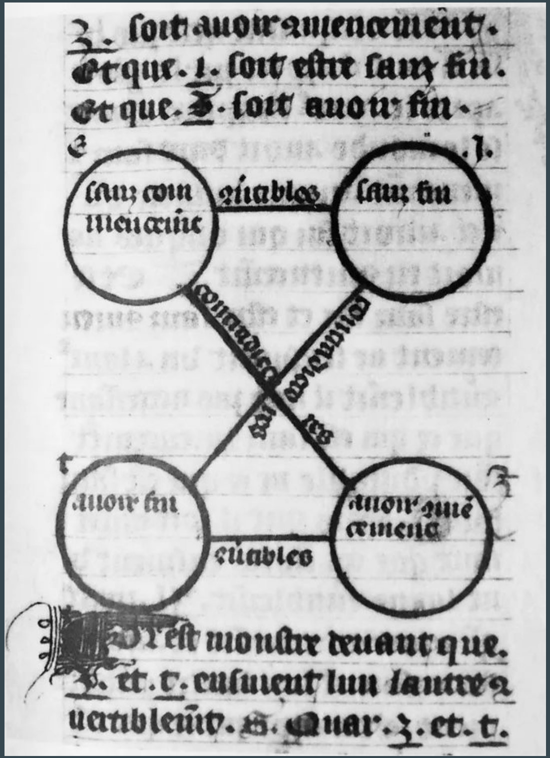
Image taken from: Borgatti, Stephen. (2005). Centrality and Network Flow. Social Networks. 27. 55-71.

Graphs and logic

"...all deductive reasoning, even simple syllogism, involves an element of observation; namely, deduction consists in constructing an icon or diagram the relations of whose parts shall present a complete analogy with those of the parts of the object of reasoning, of experimenting upon this image in the imagination, and of observing the result so as to **discover unnoticed and hidden relations** among the parts"

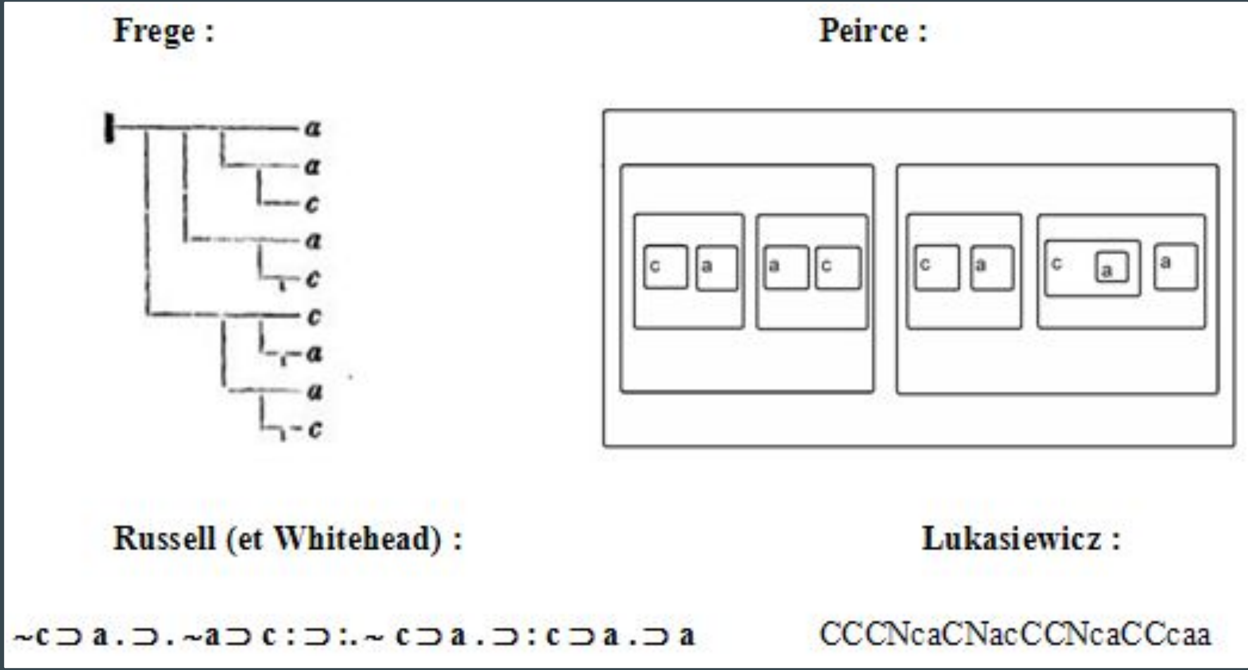
Ch. S. Peirce. *The Algebra of Logic*, 1885

Early connection of graphs and Logic



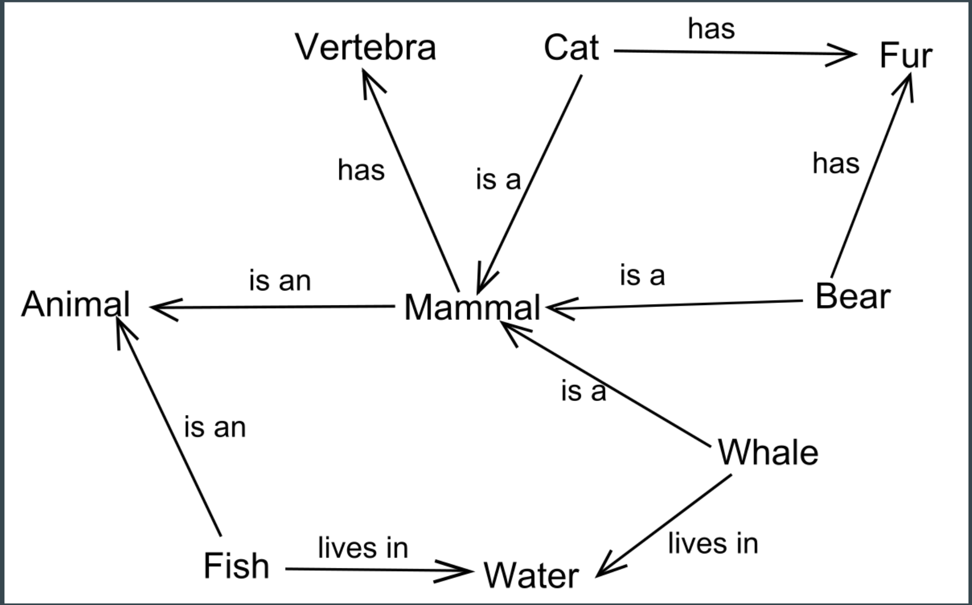
Oresme’s 14th century square of opposition.

Diagrammatic logic representations



Irving H. Anellis
https://ininet.org/how-peircean-was-the-fregean-revolution-in-logic-1.html?page=7

Semantic networks: logic specifications in a diagrammatic form



has(Mammal, Vertebra)

is_a(Cat, Mammal)

$\forall x \forall y \forall z \quad is\_a(x,y) \land has(y,z) \Rightarrow has(x,z)$

The unusual effectiveness of diagrams to represent knowledge

Fundamental ideas behind this linkage between graphs and knowledge:

- Simple way of abstracting facets of real life and processes
- Easy visualization for humans
- Simple to operationalize and communicate
- Open the possibility to find new relations that were not explicitly present in the original model or its parts

Graphs as a simple formal model for diagrams

Graphs represent a very simple and widespread conceptual model:

- ***Entities*** (represented as nodes)
- ***Relationships*** (represented as edges)

Graphs as a simple formal model for diagrams

- The model can be easily formalized in mathematical terms
- The human perception may be possible to automatize to a great extent
- It is a good idea to scale this model of reasoning beyond human capabilities

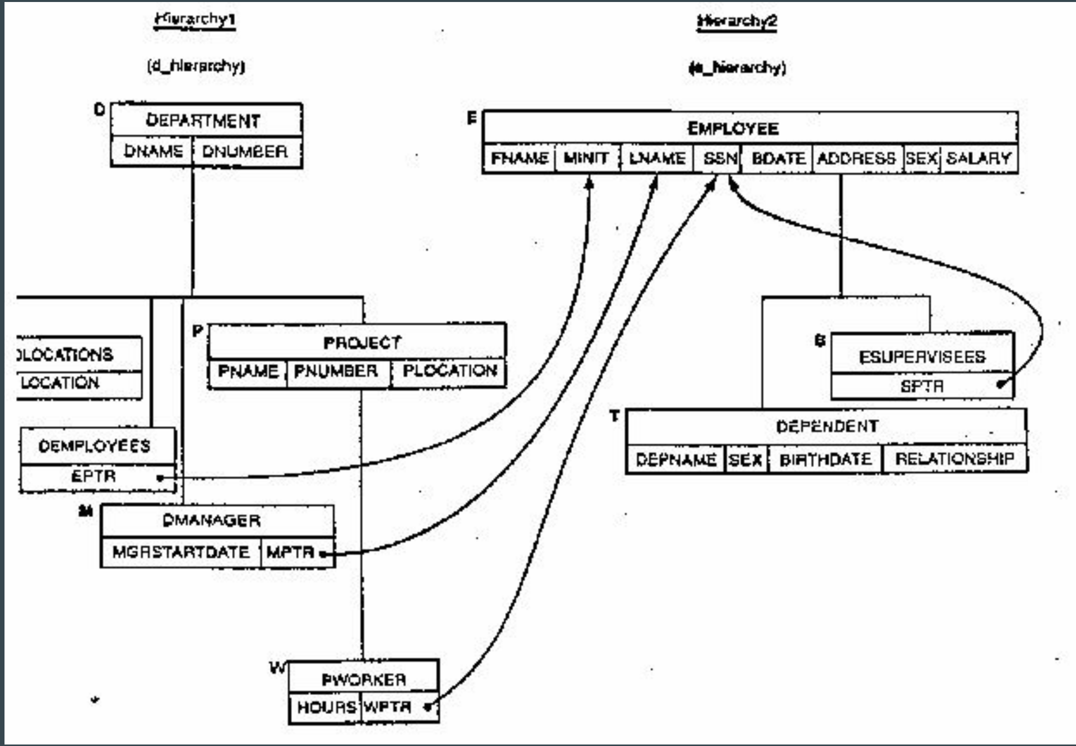
Data and graphs

Problems already discussed:

- Complex diagrams are difficult to handle in physical terms
- Many particular metadata (thus difficult to interoperate, integrate or extend)
- Paper make them static (as images)

Idea with the advent of the digital: **give flexibility to the physical representation**

A digital/data representation for graphs



"Primitive" versions:

- discrete
- formal

This led to:

- clearly defined data types
- nodes as records
- relations as pointers among registers

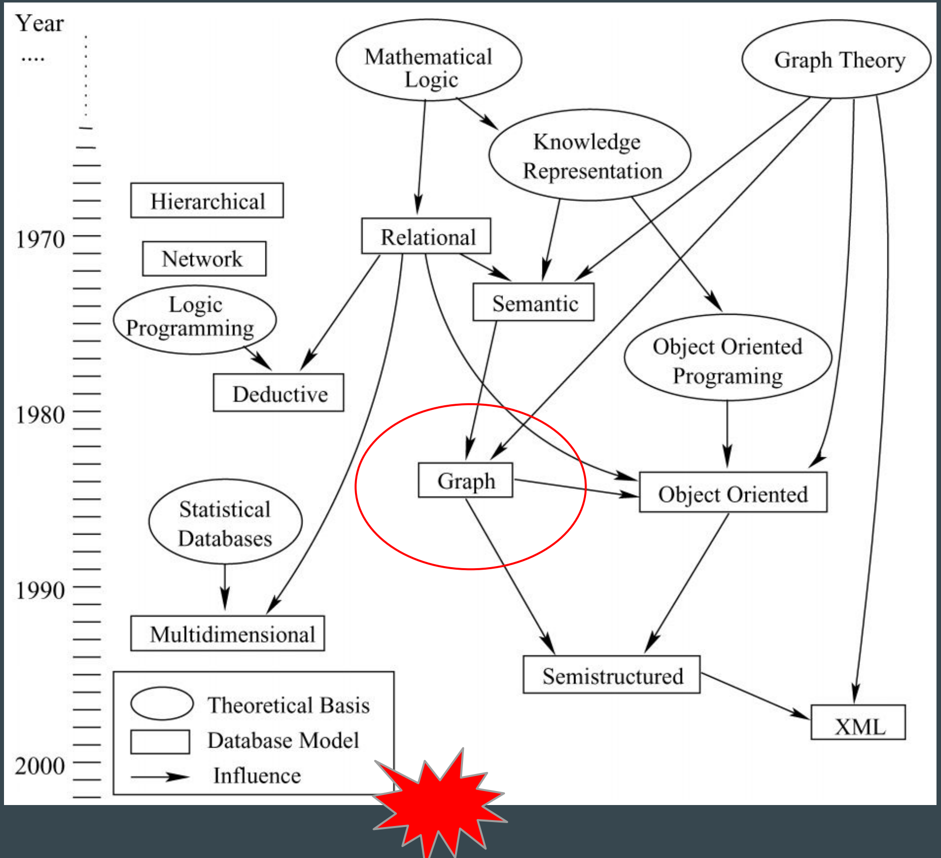
The advent of graph data models

Second wave: attempt to implement the idea of **separation of concerns**

- user view graphs
- implementation

Almost succeeded in the 1980’s: golden era of graph databases

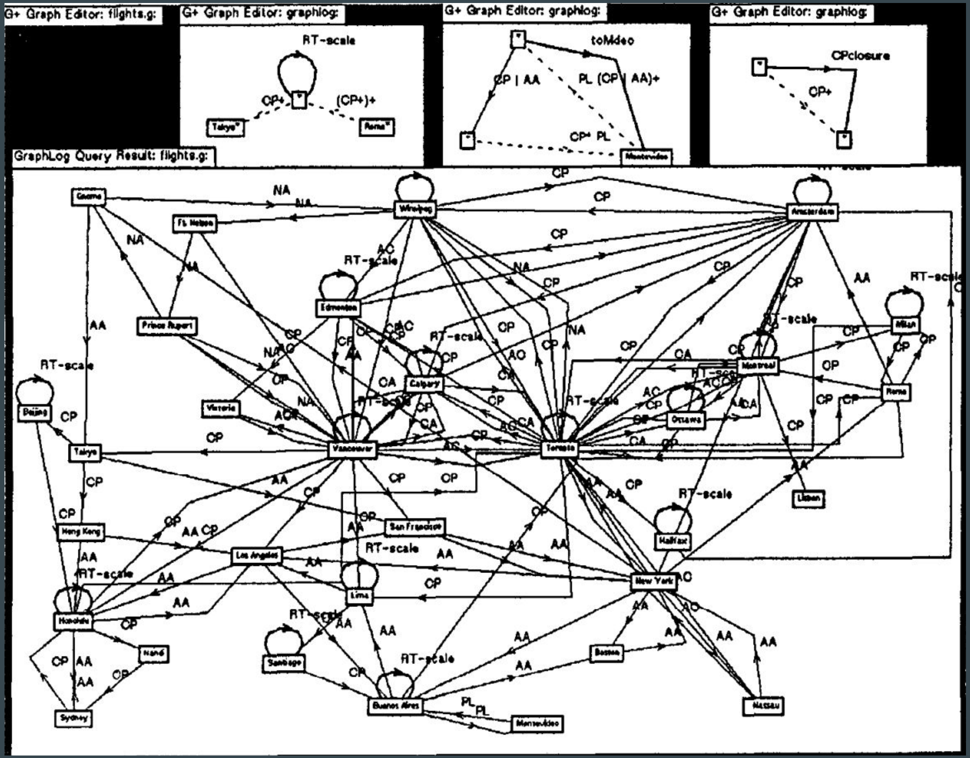
The advent of graph data models



[Diagram by A. Mendelzon
In: Angles and Gutierrez 2008]

The advent of graph data models:

Problems: hardware, software, visualization



[Consens and Mendelzon 1990]

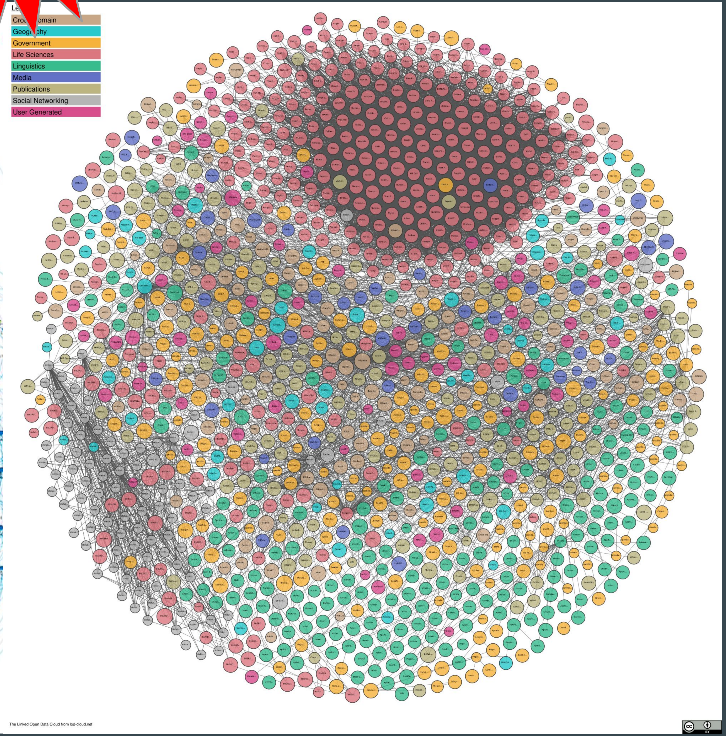
Why did graph databases become necessary?

- Scale (millions of node and edges)
  - Thus, Peirce’s insights not anymore true
- Versatile visualization software
  - Back some of Peirce’s insights
- Incorporation of different types
  - Particularly multimedia: images, sound, video

What is preserved?

- Simplicity of representation
- Simplicity of integration and extension

An upheaval to graph databases



https://medium.com/@jollyp/big-data-graph-visualisations-75f341dc36ec

https://lod-cloud.net

Graphs as mathematical structures

Extraordinary simple building blocks, and richness of representation for the construction of complex structures.

To manage appropriately large graphs we need to understand [Chung 2010]:

- What are the basic structure of such large networks?
- How do they evolve?
- What are the underlying principles that dictate their behavior?
- How are subgraphs related to the large (and often incomplete) host graphs?
- What are the main graph invariants that capture the myriad of properties of large graphs?

Some fundamental tools

The Toolbox includes [Chung 2010]:

- Combinatorial an probabilistic methods
- Spectral methods

And for non-symmetric structures:

- General random graph theory for any given degree distribution
- Percolation in general host graphs
- PageRank for representing quantitative correlations
- Game aspects

The geometry of graphs

Three main types of properties can be distinguished in graphs:

- **Local properties:** nodes and neighborhoods
- **Connectivity properties:** paths and their regular and logical expressions
- **Global properties:** networks analysis, analytics in general

Querying:
What are new challenges?
What are new techniques?

…

A conceptual view of querying graphs

- **Local properties**

  Extracting nodes from a graph: first-order logic with bounded resources and graph neural networks

- **Connectivity properties**

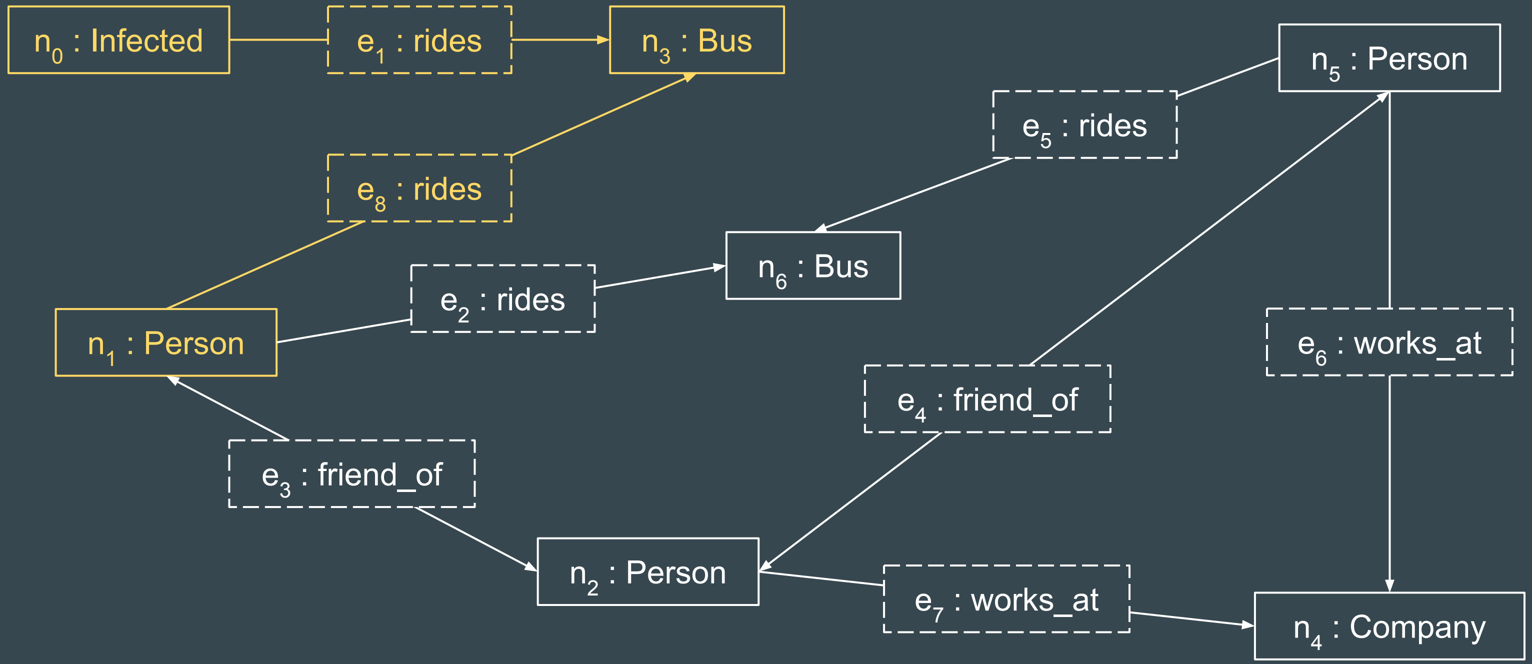
  Extracting paths from a graph: approximation and uniform generation
  Paths as first-class citizens

- **Global properties**

  Explainable AI and the search of a declarative language for interpretability

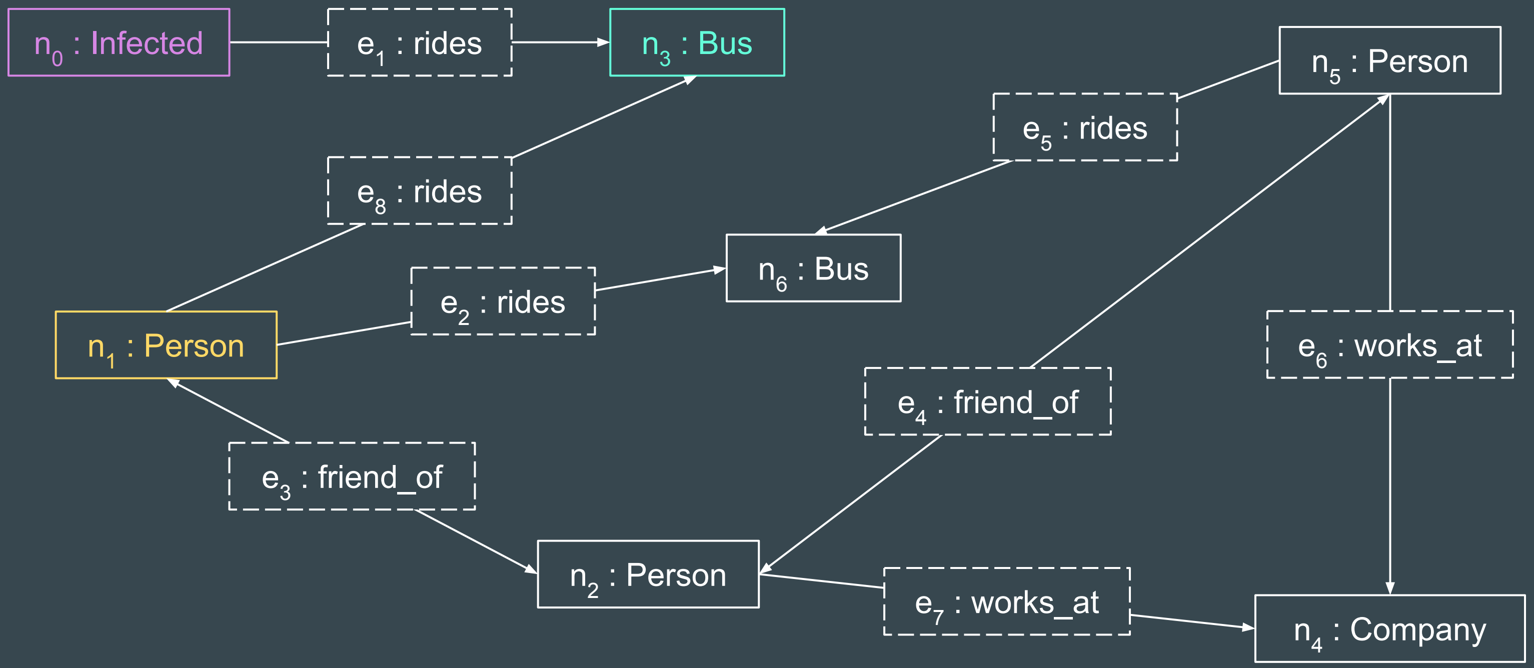
Extracting nodes from a graph: an old problem

The notion of close contact: ?Person/rides/?Bus/**rides⁻**/?Infected



An old idea: use first order-logic (FO) as a query language

$$\text{Person}(\mathbf{x}) \wedge \exists \mathbf{y}\, [\text{rides}(\mathbf{x},\mathbf{y}) \wedge \text{Bus}(\mathbf{y}) \wedge \exists \mathbf{z}\, (\text{rides}(\mathbf{z},\mathbf{y}) \wedge \text{Infected}(\mathbf{z}))]$$



Also an old idea: use FO with bounded resources

$\text{Person}(x) \wedge \exists y\,[\text{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\,(\text{rides}(x,y) \wedge \text{Infected}(x))]$

Only two variables are needed

- Only the values of these variables need to be stored
- No need to store partial results from joins of arbitrary size

Evaluating FO with bounded resources

$\text{Person}(x) \wedge \exists y\, [\text{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\, (\text{rides}(x,y) \wedge \text{Infected}(x))]$

Person
x
n_1
n_2
n_6

rides	
x	y
n_0	n_3
n_1	n_3
n_1	n_6
n_5	n_6

Bus
y
n_3
n_6

rides	
x	y
n_0	n_3
n_1	n_3
n_1	n_6
n_5	n_6

Infected
x
n_0

Evaluating FO with bounded resources

$Person(x) \wedge \exists y\, [rides(x,y) \wedge Bus(y) \wedge \exists x\, (rides(x,y) \wedge Infected(x))]$

Person
x
n_1
n_2
n_6

rides	
x	y
n_0	n_3
n_1	n_3
n_1	n_6
n_5	n_6

Bus
y
n_3
n_6

x	y
n_0	n_3

Evaluating FO with bounded resources

$Person(x) \land \exists y\, [rides(x,y) \land Bus(y) \land \exists x\, (rides(x,y) \land Infected(x))]$

Person
x
n_1
n_2
n_6

rides	
x	y
n_0	n_3
n_1	n_3
n_1	n_6
n_5	n_6

Bus
y
n_3
n_6

y
n_3

Evaluating FO with bounded resources

$\text{Person}(x) \wedge \exists y\, [\text{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\, (\text{rides}(x,y) \wedge \text{Infected}(x))]$

Person
x
n_1
n_2
n_6

rides	
x	y
n_0	n_3
n_1	n_3
n_1	n_6
n_5	n_6

y
n_3

Evaluating FO with bounded resources

$\text{Person}(x) \land \exists y\, [\text{rides}(x,y) \land \text{Bus}(y) \land \exists x\, (\text{rides}(x,y) \land \text{Infected}(x))]$

Person
x
n_1
n_2
n_6

x	y
n_0	n_3
n_1	n_3

Evaluating FO with bounded resources

Person(x) ∧ ∃y [rides(x,y) ∧ Bus(y) ∧ ∃x (rides(x,y) ∧ Infected(x))]

Person
x
n_1
n_2
n_6

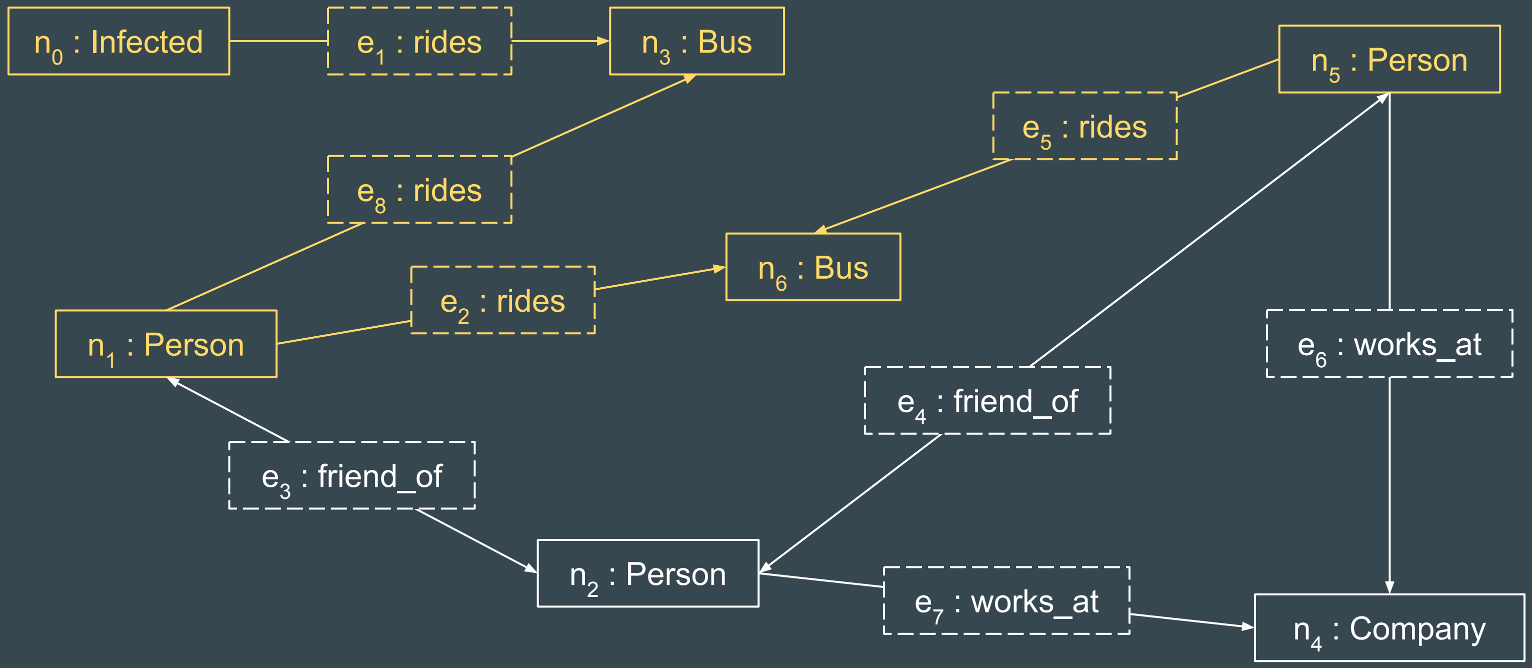
x
n_0
n_1

Evaluating FO with bounded resources

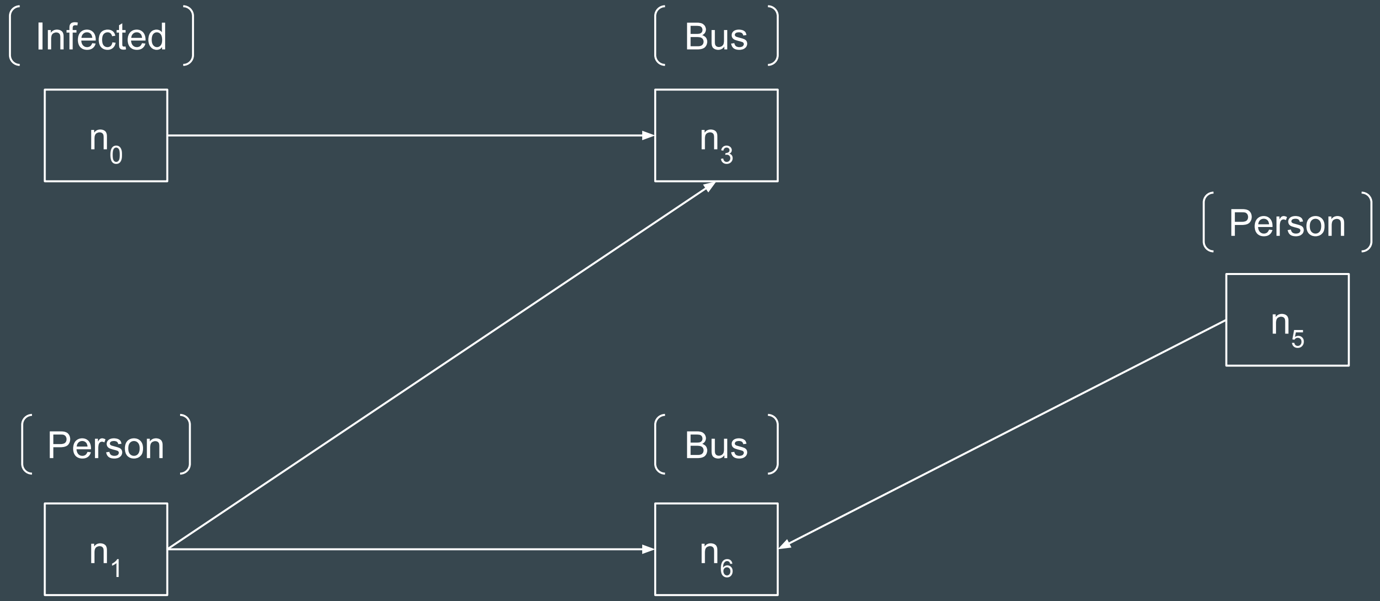
$$\text{Person}(x) \land \exists y\, [\text{rides}(x,y) \land \text{Bus}(y) \land \exists x\, (\text{rides}(x,y) \land \text{Infected}(x))]$$

x
n_1

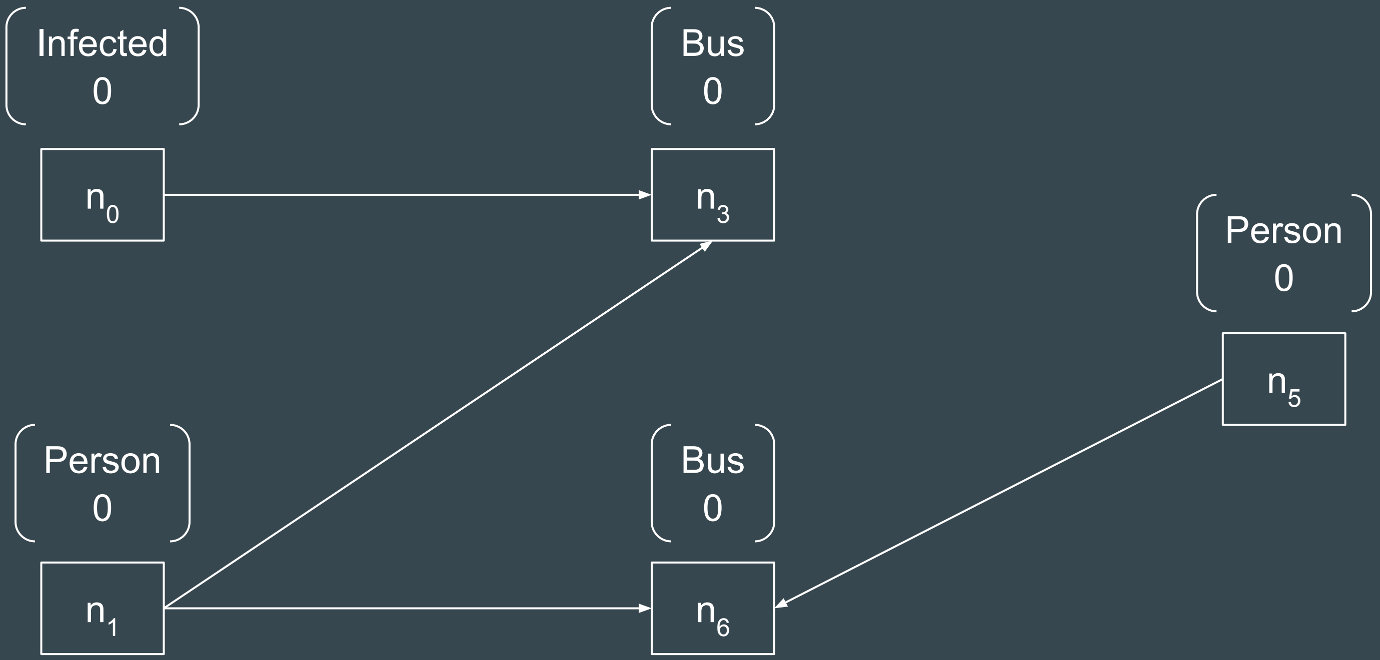
On the other side: Graph neural networks (GNNs)



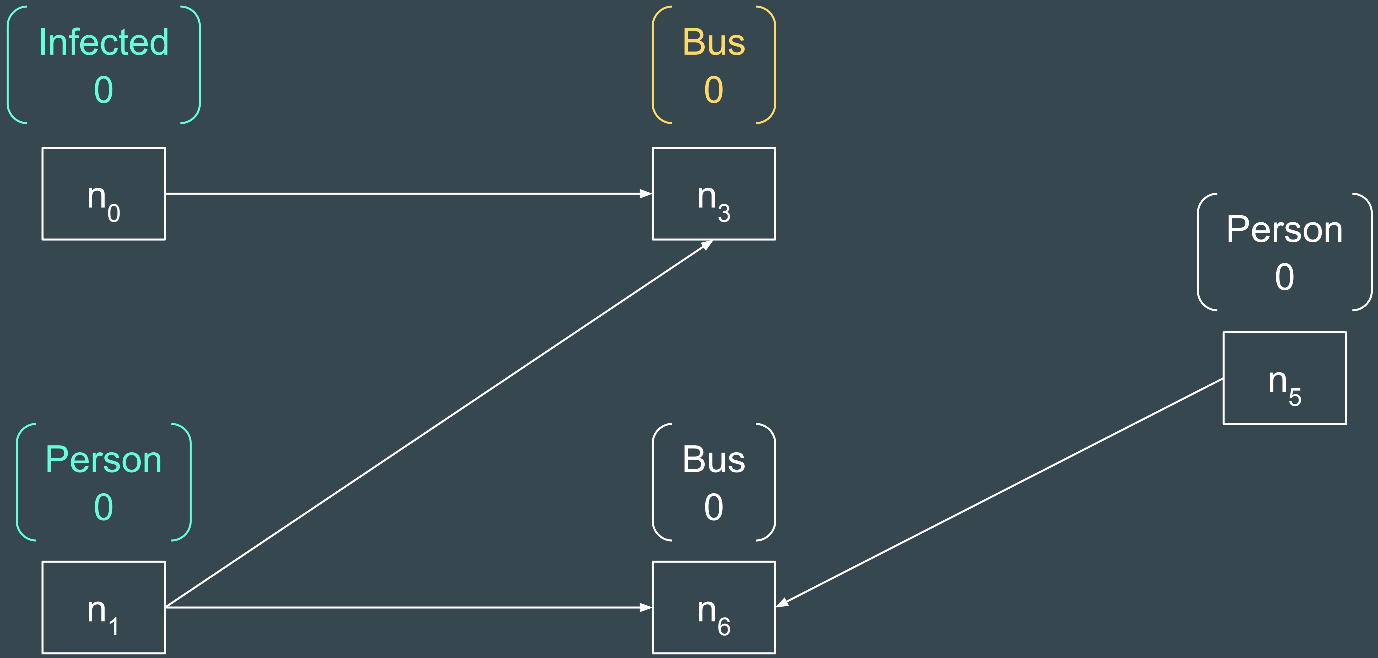
On the other side: Graph neural networks (GNNs)



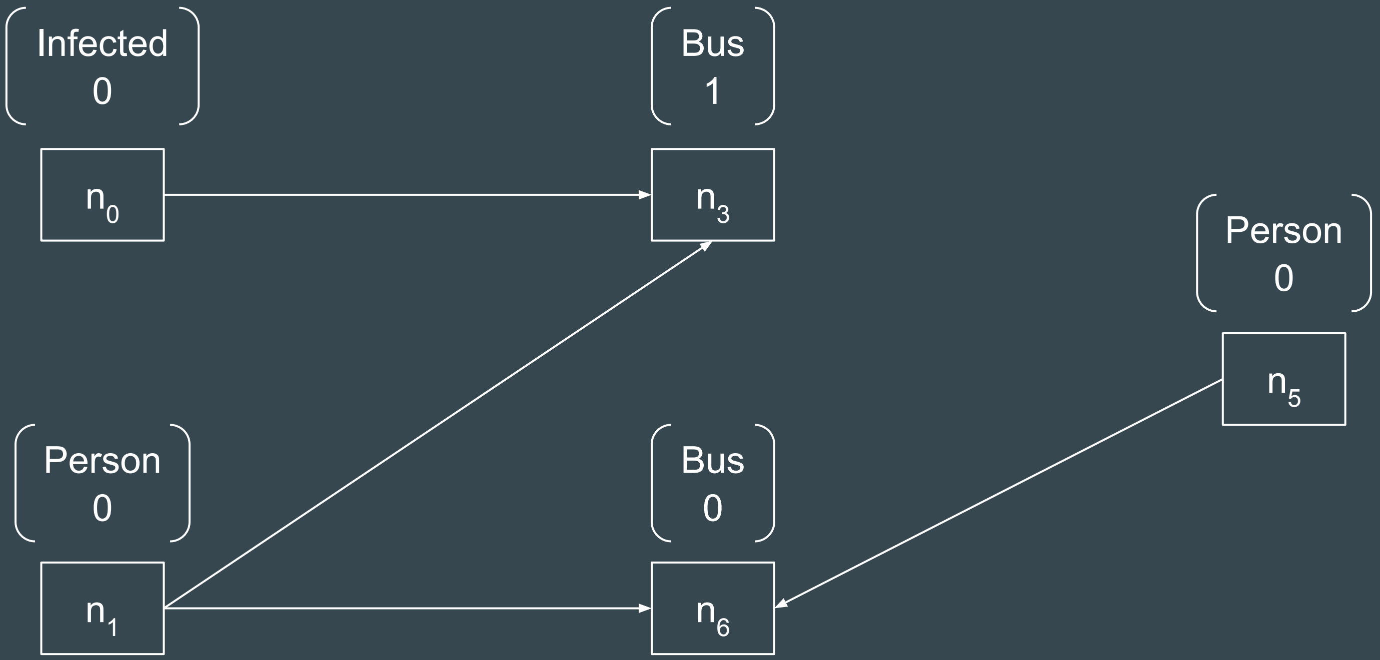
Processing by layers in GNNs: the input



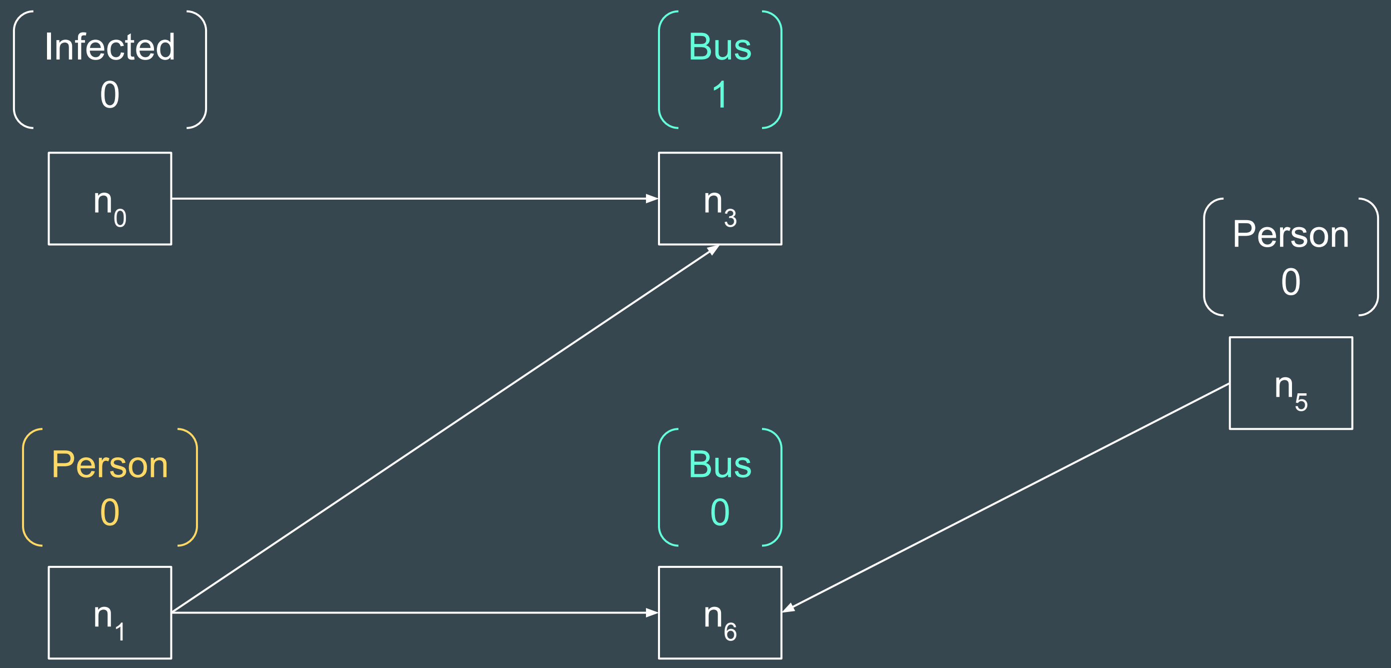
Computing the first layer



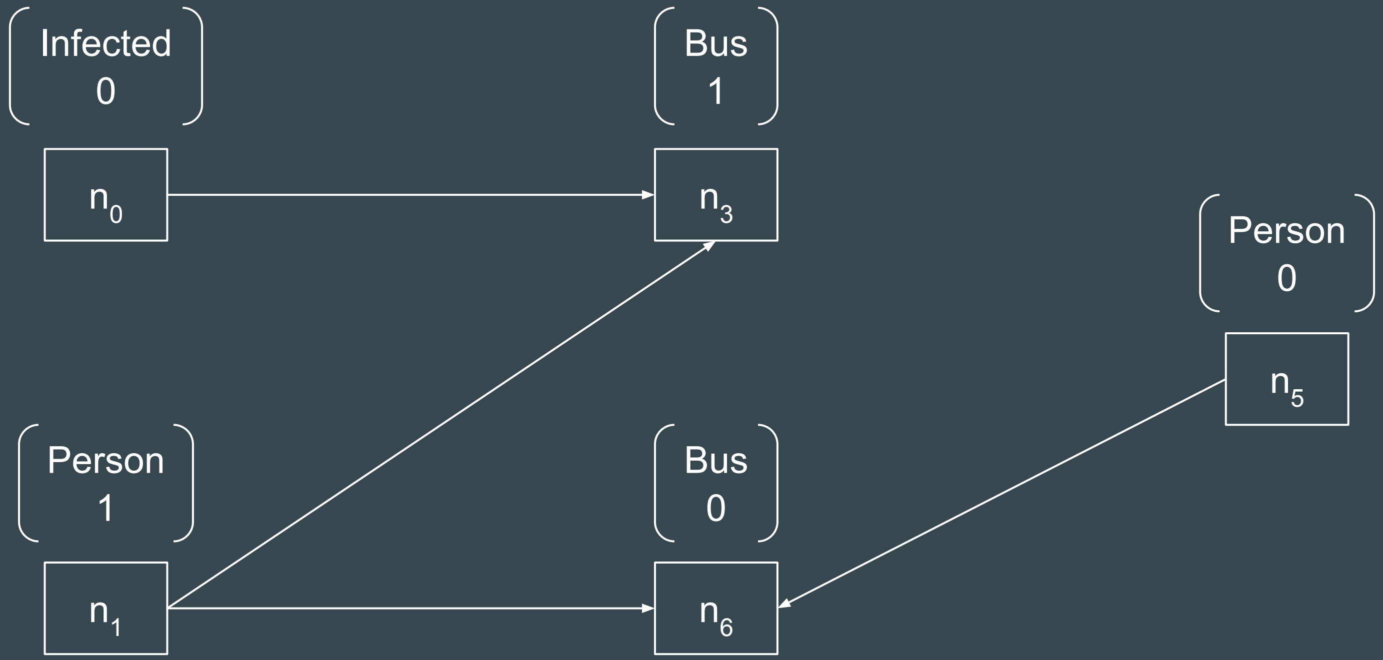
The result of the first layer



Computing the next layer



The result of second layer



The architecture of GNNs

$u^{(i)}$: vector of features of node u at layer i

- $u^{(0)}$ is the vector of features from the input graph

$u^{(i+1)}$ = **COMB**($u^{(i)}$, **AGG**({{ $v^{(i)}$ | u and v are neighbors in G }}))

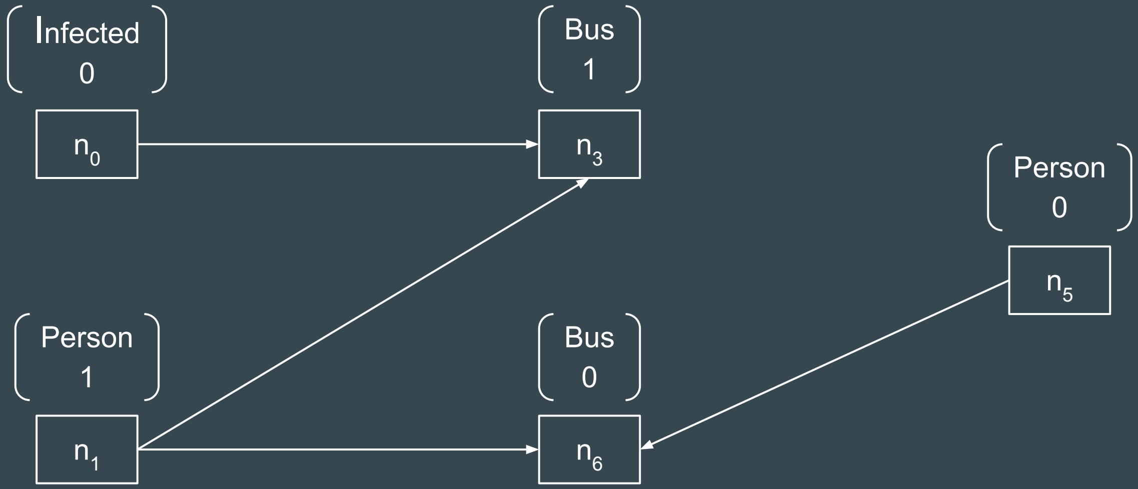
If k is the last layer: **CSL**($u^{(k)}$) is the result for node u

The architecture of GNNs

?Person/rides/?Bus/rides⁻/?Infected

$$\textbf{CSL}\begin{pmatrix}\text{Person}\\ 1\end{pmatrix} = 1$$

$$\textbf{CSL}\begin{pmatrix}\text{Person}\\ 0\end{pmatrix} = 0$$



How are the previous paradigms related?

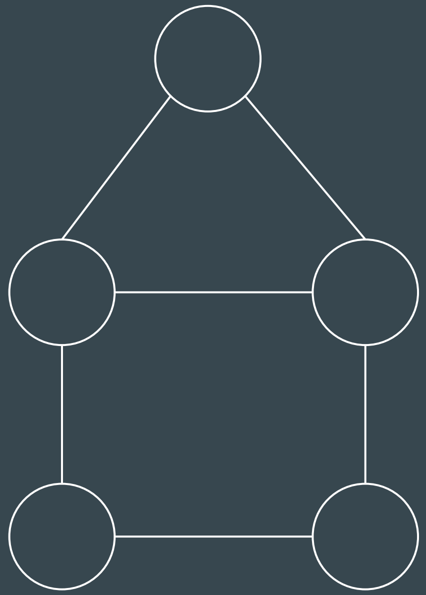
A new idea: the Weisfeiler-Lehman (WL) graph isomorphism test makes the connection

Two alternative points of view:

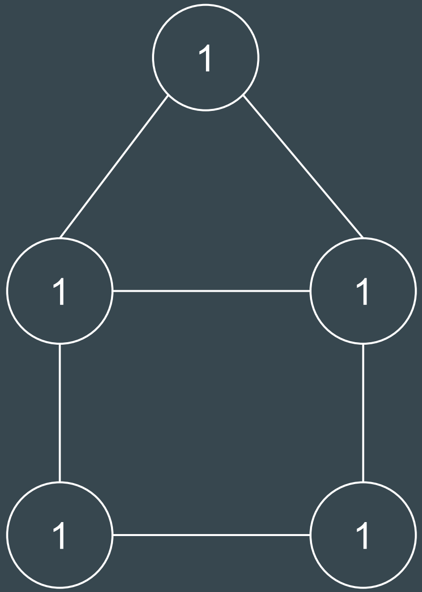
- First-order logic is a declarative query language, with a well-known and studied inference mechanism
- GNNs are a popular classification paradigm, with a growing number of algorithms and techniques to *learn* and implement them

The bridge: WL test for graph isomorphism

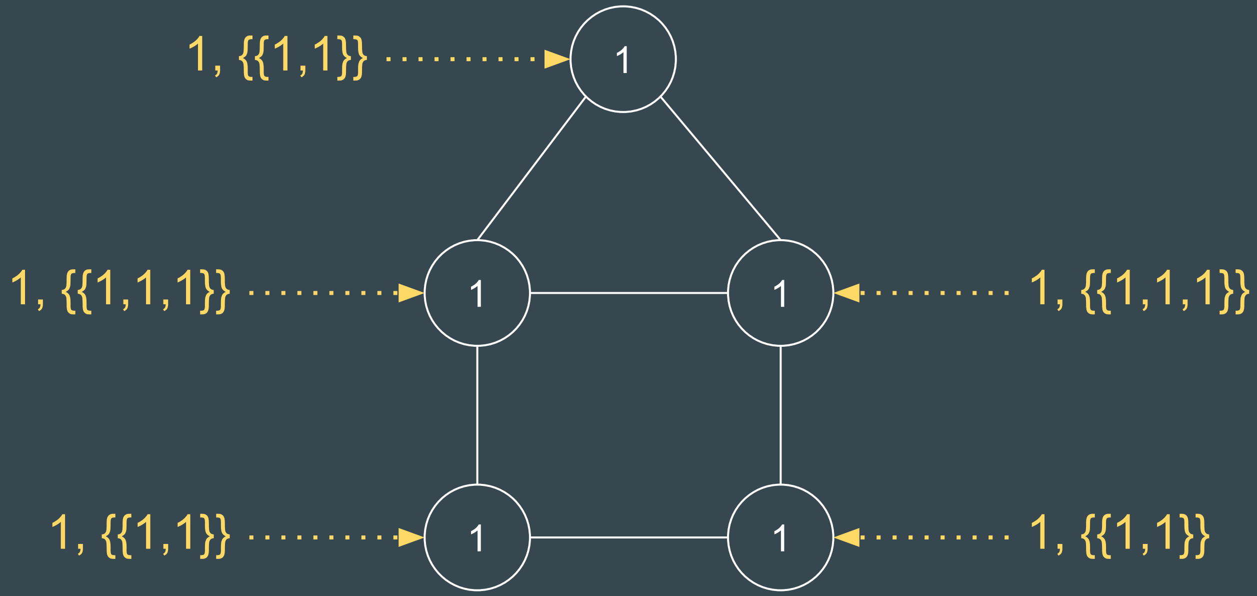
We will construct a canonical representation
(1-dimensional test)



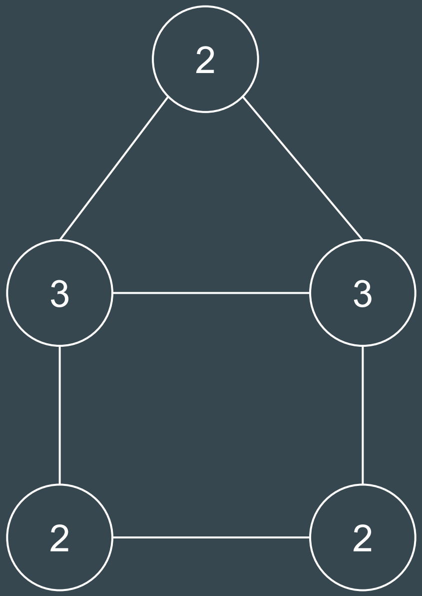
The WL test for graph isomorphism



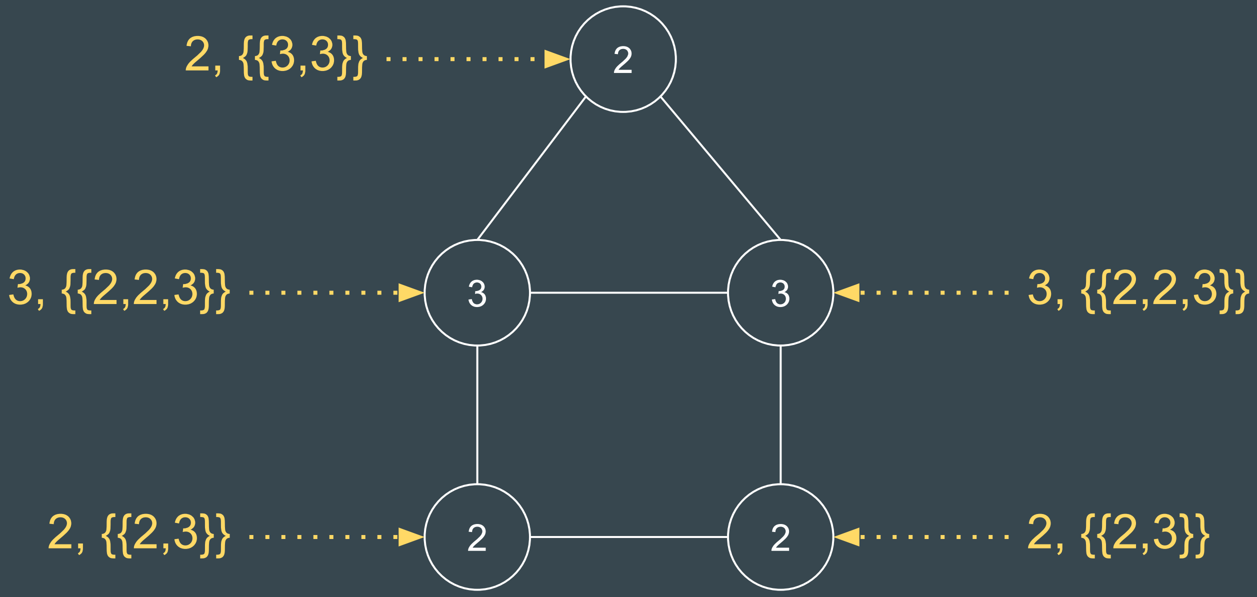
The WL test for graph isomorphism



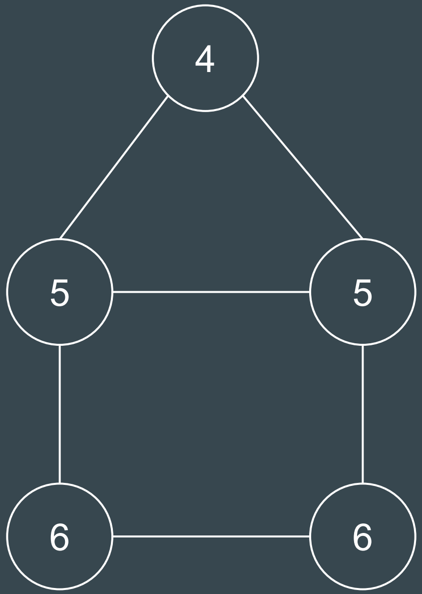
The WL test for graph isomorphism



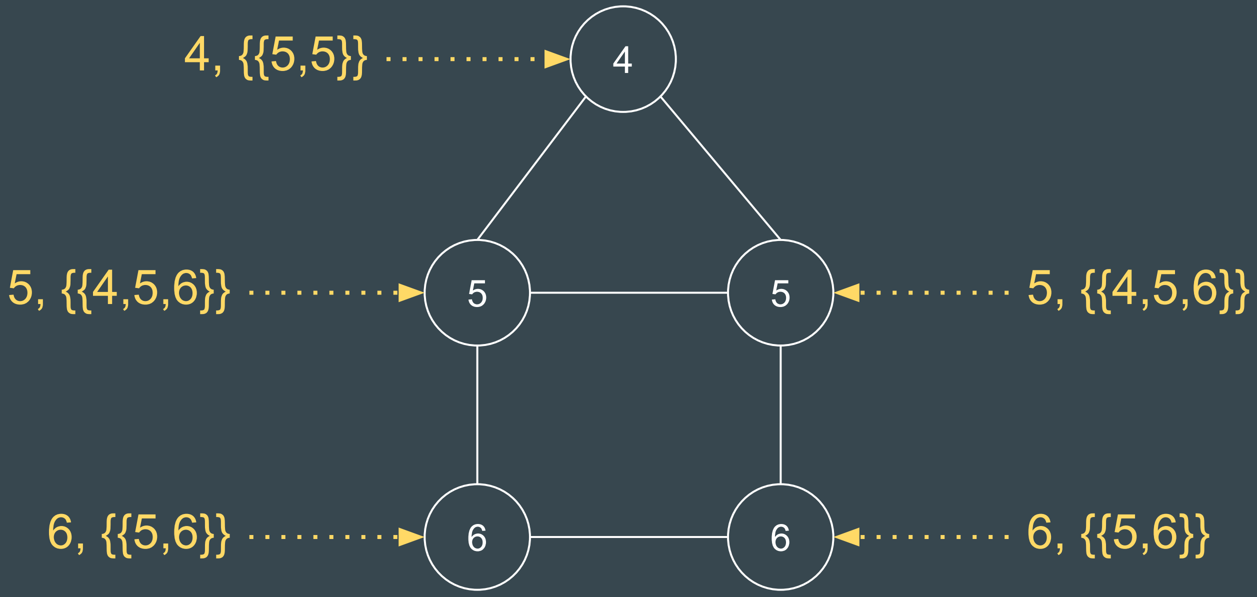
The WL test for graph isomorphism



The WL test for graph isomorphism

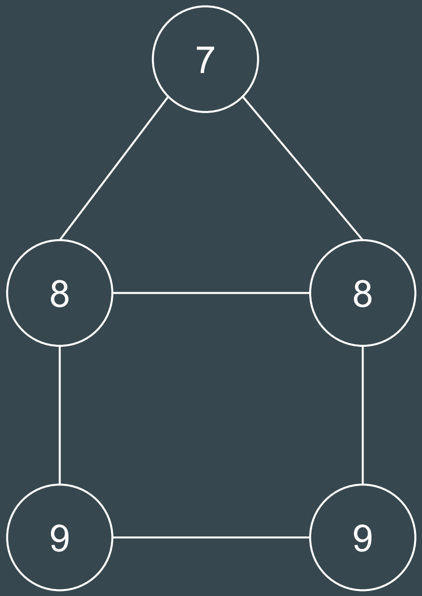


The WL test for graph isomorphism



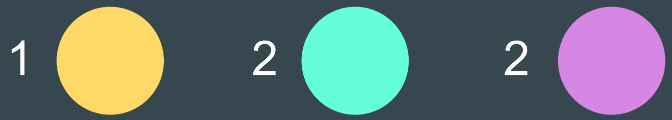
The WL test for graph isomorphism

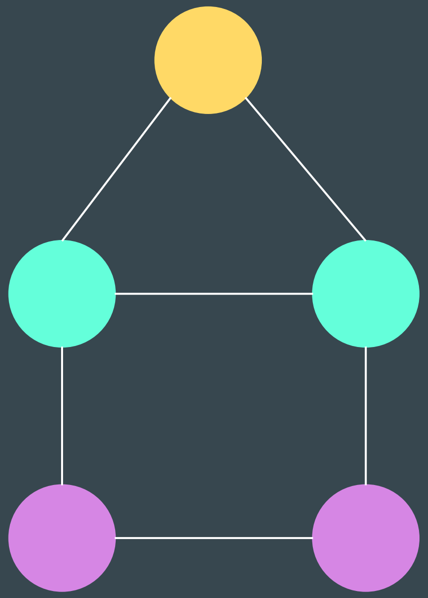
We reach a fixpoint



The WL test for graph isomorphism

The Canonical representation
(1-dimensional test)





The WL test and GNNs

The WL test can be considered as a heuristic to verify whether two graph are isomorphic

- But a very good heuristic, with theoretical guarantees

The WL test obviously resembles the way in which GNNs work

Theorem [Morris et al. 2019, Xu et al. 2019]: If WL assigns the same color to u and v in a graph G, then every (aggregate-combine) GNN classifies u and v in the same way on input G

The WL test and FO

Consider the fragment FOC^2 of FO:

$\text{Person}(\mathbf{x}) \wedge \exists y\, [\text{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists \mathbf{x}\, (\text{rides}(x,y) \wedge \text{Infected}(x))]$

$\text{Person}(x) \wedge \exists^{\geq 2} \mathbf{y}\, [\text{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\, (\text{rides}(x,y) \wedge \text{Infected}(x))]$

FOC^2 can be efficiently evaluated as shown before (using only binary tables)

Theorem [Cai et at. 1992]: WL assigns the same color to u and v in a graph G if and only if either $u,v \in Q(G)$ or $u,v \notin Q(G)$, for every unary query $Q(x)$ in FOC^2

Putting all together

Theorem [Barceló et al. 2020]: There exists a (natural) fragment of FOC^2 with the same expressive power as (aggregate-combine) GNNs

Such a fragment of FOC^2 includes the previous formulae:

$$\text{Person}(x) \wedge \exists y\, [\textbf{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\, (\textbf{rides}(x,y) \wedge \text{Infected}(x))]$$

$$\text{Person}(x) \wedge \exists^{\geq 2} y\, [\textbf{rides}(x,y) \wedge \text{Bus}(y) \wedge \exists x\, (\textbf{rides}(x,y) \wedge \text{Infected}(x))]$$

Some questions to think about

Can learning techniques for GNNs be used for learning queries on graphs?

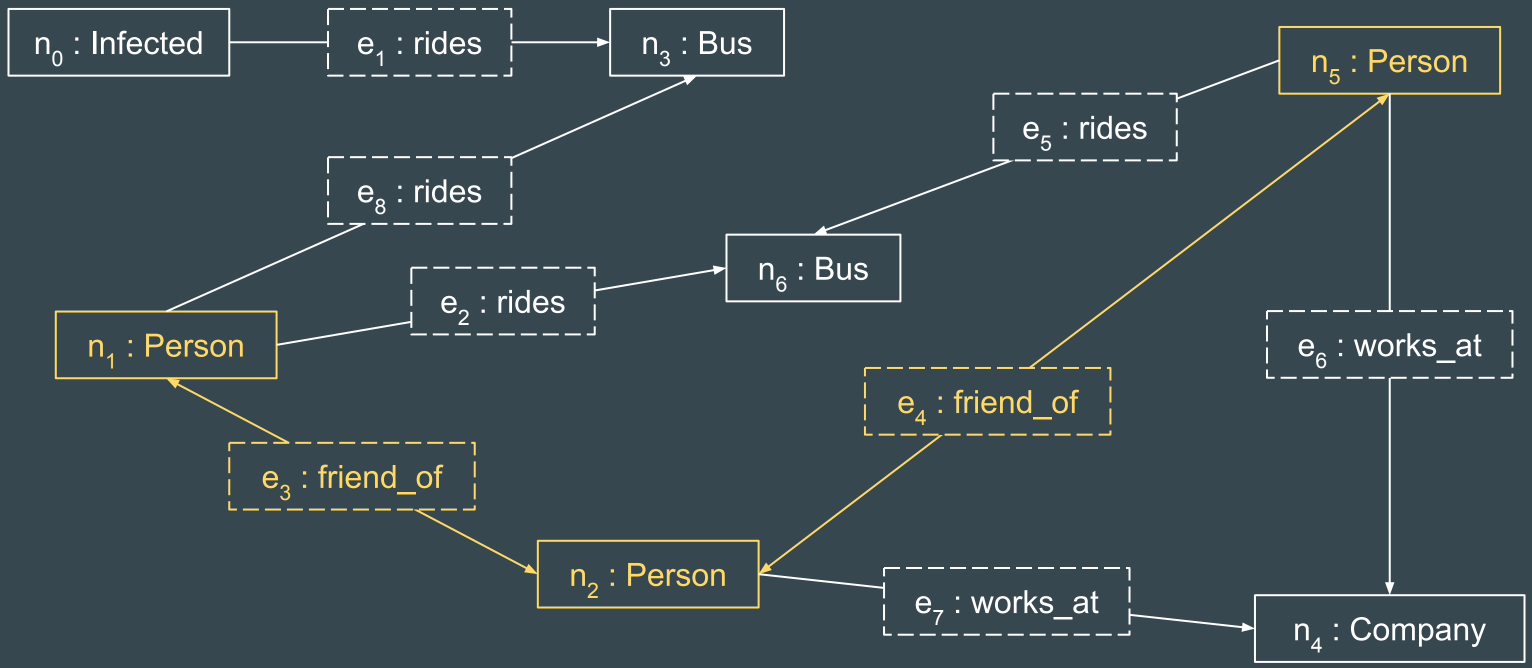
- What are good algorithms for translating (aggregate-combine) GNNs into (well-studied) declarative query languages?

What is the appropriate GNN architecture for regular expressions?

- A form of recursion need to be included

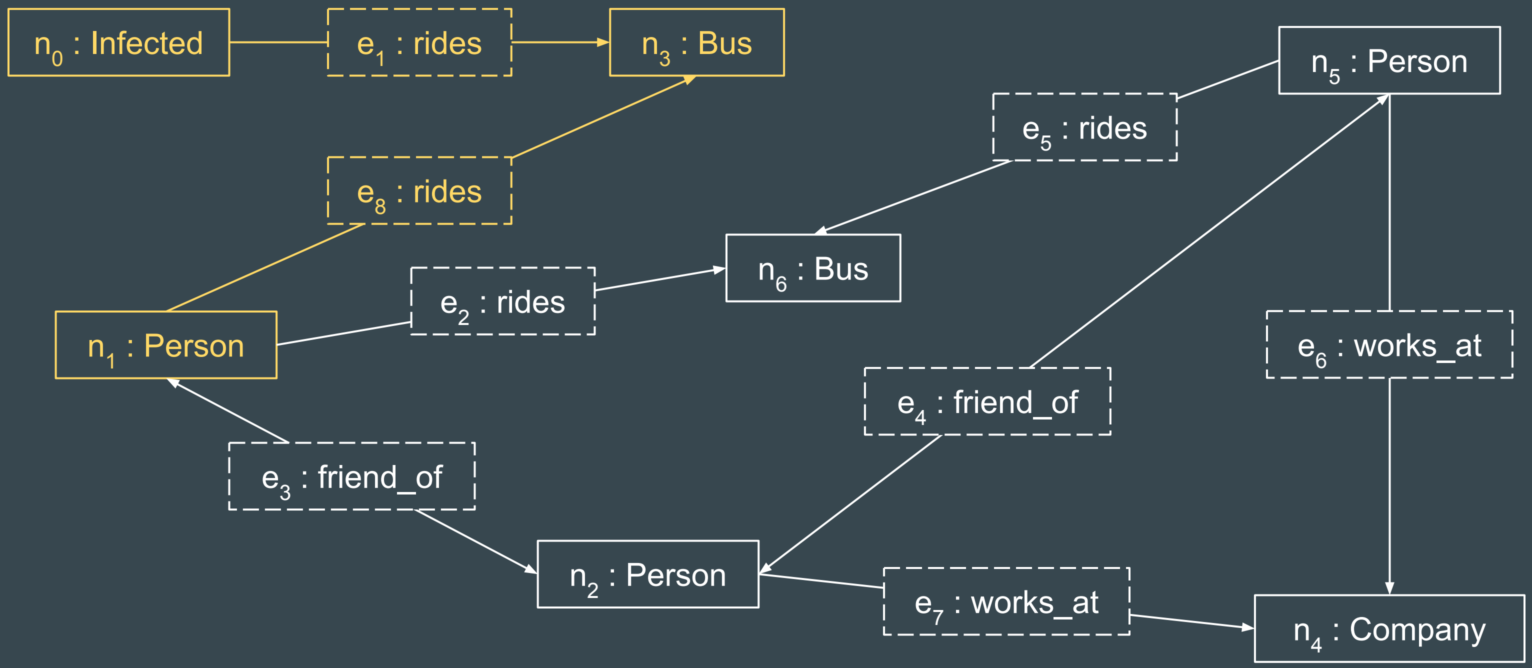
Extracting paths from a graph: also an old problem

A canonical example: ?Person/(friend_of/?Person)+



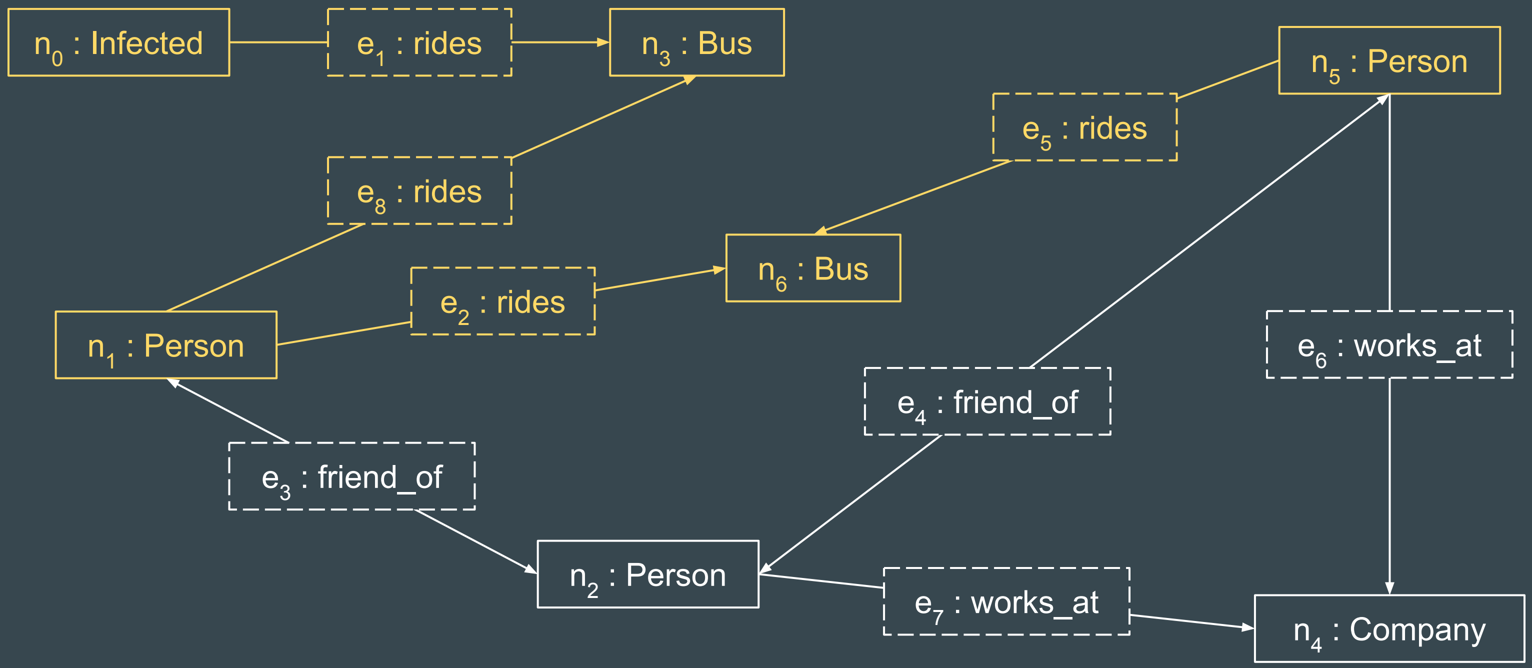
Extracting paths from a graph: also an old problem

The notion of close contact: ?Infected/rides/?Bus/rides⁻/?Person



Extracting paths from a graph: also an old problem

A stricter notion of close contact: ?Infected/(rides/?Bus/rides⁻/?Person)+



A new approach to an old problem

Give up completeness: we do not want to find *all* paths that conforms to a regular expression, even if their length is given as parameter

We consider problems:

- **COUNT(G, r, n):** count the number of paths p in G such that p conforms to r and the length of p is n
- **GEN(G, r, n):** generate uniformly at random a path p in G such that p conforms to r and the length of p is n

Are these difficult problems?

Without including regular expressions as parameter, COUNT(G, n) can be solved efficiently by a dynamic programming approach

COUNT(G, r, n) is #P-complete

- If it can be solved in polynomial time, then P = NP

How do we solve the previous problems? Give up precision

Randomized approximation to the rescue

#P-hardness of COUNT(G, r, n) does not preclude the existence of an approximation algorithm for this problem

We would like to have an algorithm A(G, r, n, **ε**) that approximates COUNT(G, r, n) with a relative error **ε**

- It should run in time polynomial in |G| + |r| + n and in 1/**ε**

Randomized approximation to the rescue

But we would also need randomization in this case.

We ask A(G, r, n, **ε**) to be a fully polynomial-time randomized approximation scheme:

$$\Pr\left(\left| \frac{\text{COUNT}(G, r, n) - A(G, r, n, \boldsymbol{\varepsilon})}{\text{COUNT}(G, r, n)} \right| \leq \boldsymbol{\varepsilon} \right) \geq 1 - \frac{1}{2^{100}}$$

and A(G, r, n, **ε**) runs in time poly(|G|, |r|, n, 1/**ε**)

Randomized approximation to the rescue

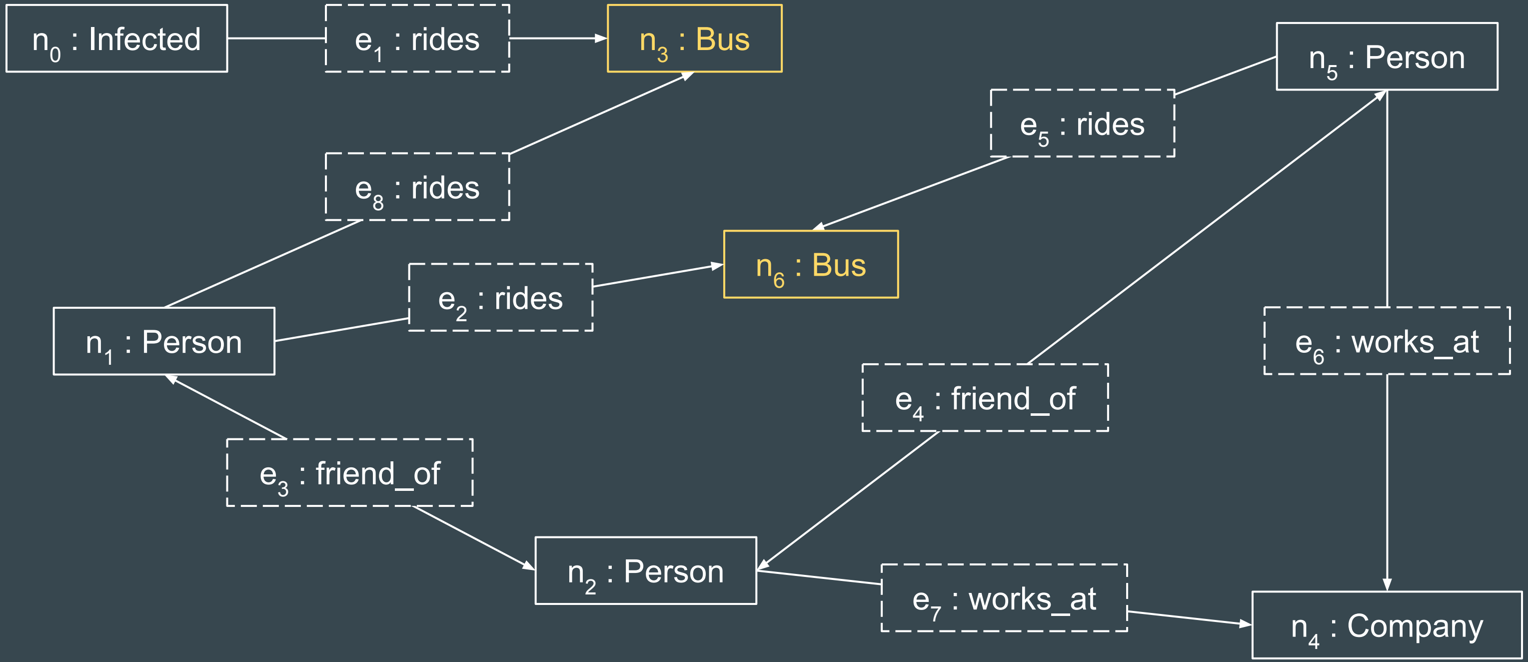
Theorem [Arenas et al. 2019]: There exists a fully polynomial-time randomized approximation scheme for COUNT(G, r, n)

Such a schema can be used to provide a randomized algorithm for GEN(G, r, n)

- Samples can be generated efficiently with an *almost* uniform distribution

An application to global properties

How important is a bus service?



Betweenness centrality of a node in a graph

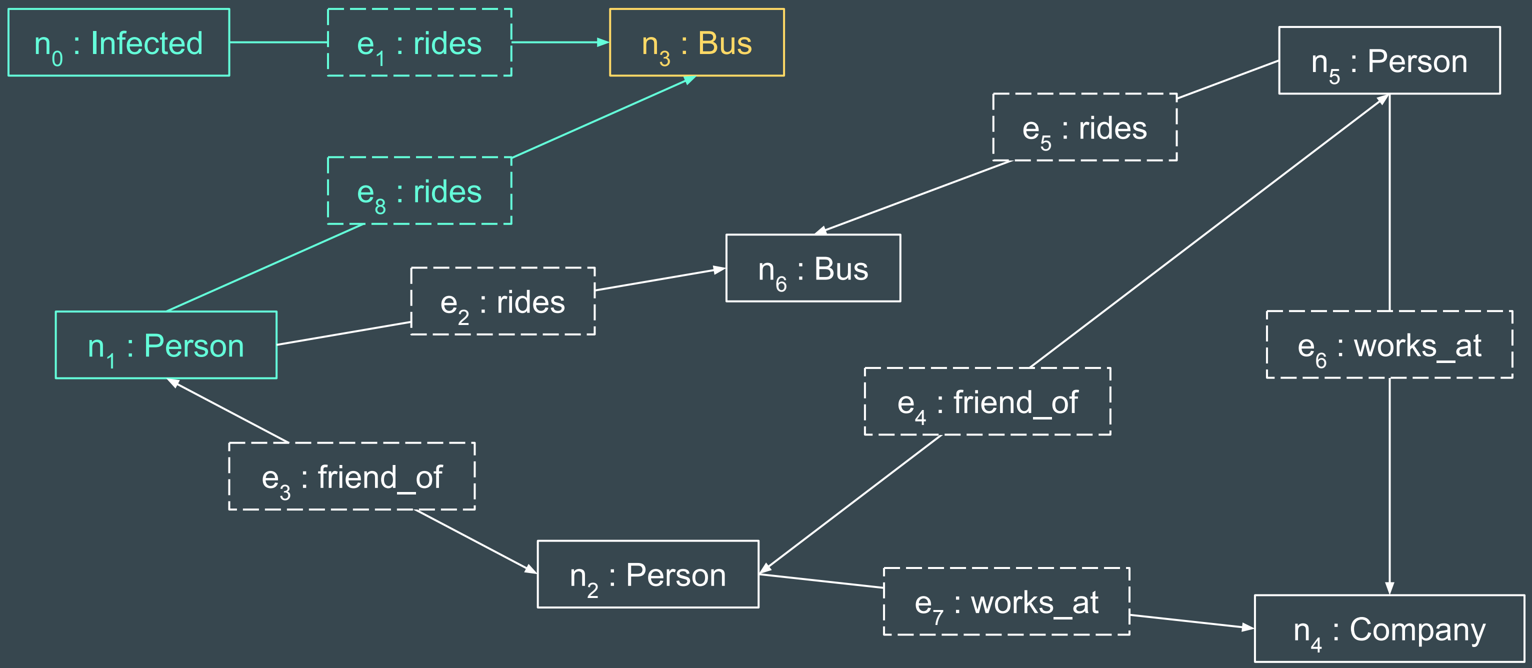
- $S_{a,b}$: set of shortest paths from a to b in G
- $S_{a,b}(u)$: set of paths in $S_{a,b}$ that include node u

$$bc(u) = \sum_{\substack{a,\, b \\ a \neq u \;\wedge\; b \neq u}} \frac{|S_{a,b}(u)|}{|S_{a,b}|}$$

Betweenness centrality can be computed in polynomial time.

Is betweenness centrality the appropriate notion?

How important is a bus service in the spread of a communicable disease?



Restricting paths in betweenness centrality

- $S_{a,b,r}$: set of shortest paths from a to b in G that conforms to regular expression r
- $S_{a,b,r}(u)$: set of paths in $S_{a,b,r}$ that include node u

$$bc_r(u) = \sum_{\substack{a,\, b \\ a \neq u \,\wedge\, b \neq u}} \frac{|S_{a,b,r}(u)|}{|S_{a,b,r}|}$$

Can this notion of centrality be computed in polynomial time?

Randomized approximation again to the rescue

Use previous algorithm to approximate $|S_{a,b,r}(u)|$ and $|S_{a,b,r}|$

- For example, with errors $\boldsymbol{\varepsilon}^2/4$ and $\boldsymbol{\varepsilon}/4$ for $|S_{a,b,r}(u)|$ and $|S_{a,b,r}|$, respectively

The you get an approximation of $bc_r(u)$ with error $\boldsymbol{\varepsilon}$ that can be computed in polynomial time.

Paths should be treated as first-class citizens

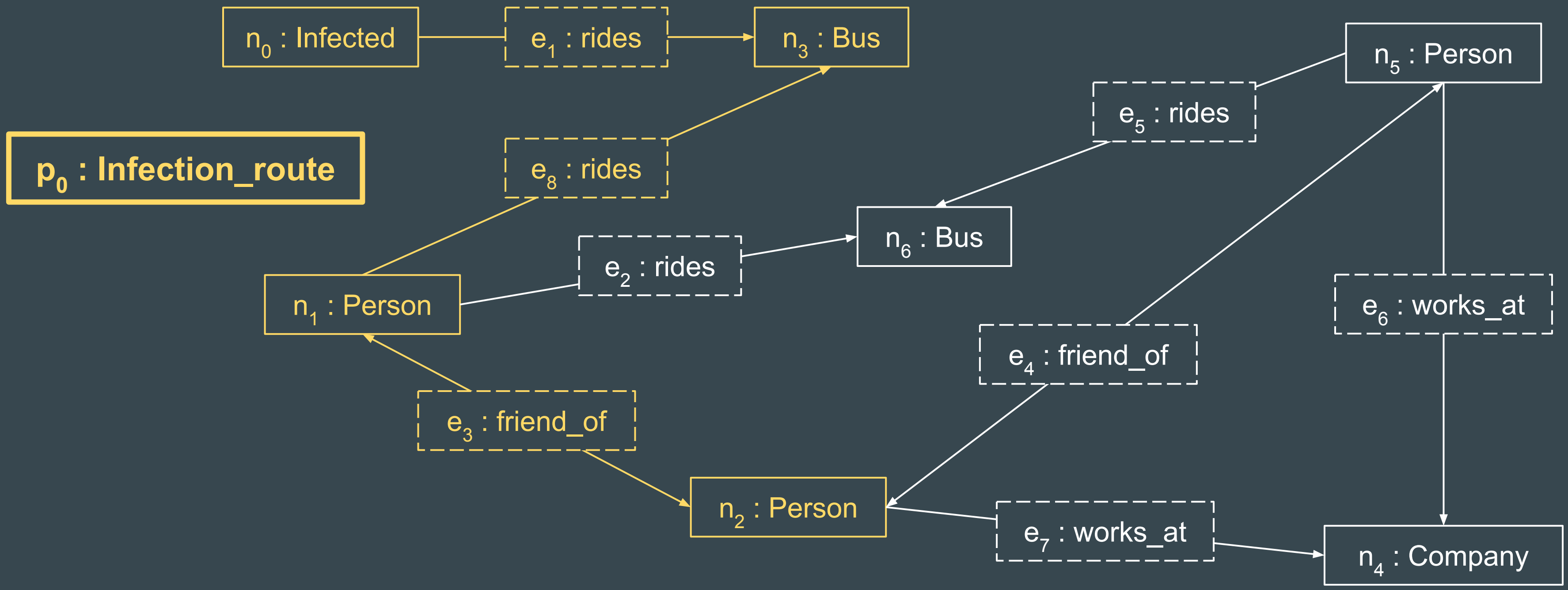
Treated at the same level as nodes and edges, so that paths can

- be materialized and stored
- have labels
- have values for properties, or associated vectors of features

The previous centrality measure can be formulated as a query over a set of paths

- Such set of paths can be defined by a sub-query

Paths as first-class citizens



Some questions to think about

Can a fully polynomial-time randomized approximation scheme for COUNT be effectively used in practice?

- Can be used to provide *fair* answers in practice?

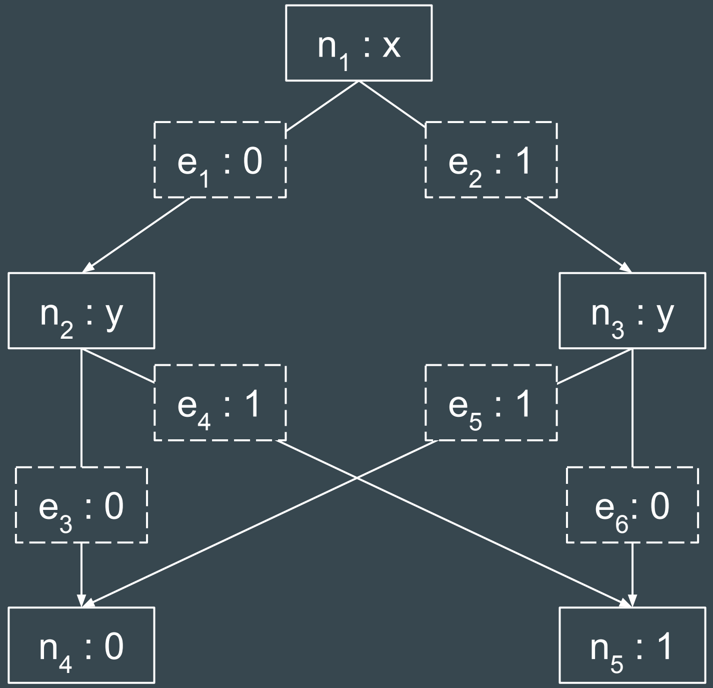
How can centrality measures be adapted to deal with knowledge graphs?

- What is an appropriate definition of a centrality measure that takes labels into account?

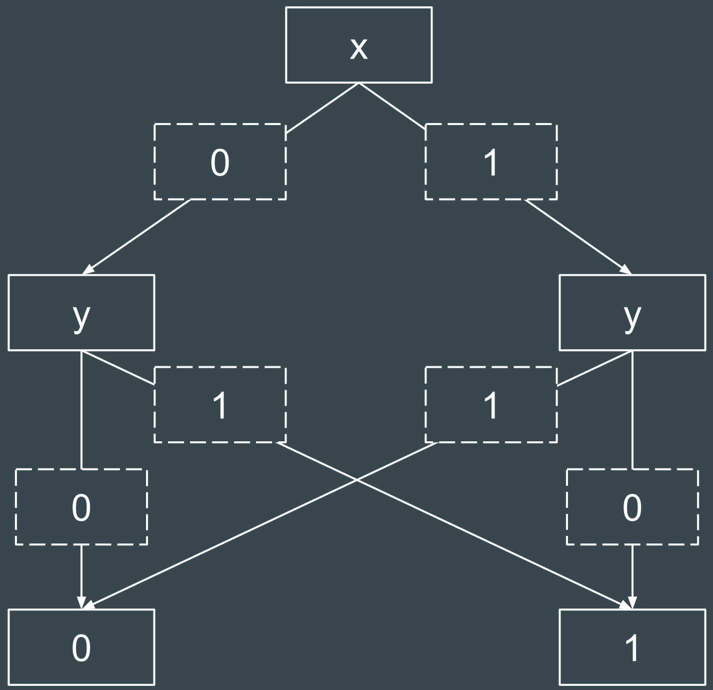
How can paths be included as first class citizens in a query language?

- A proposal in the query language G-CORE introduced in [Angles et al. 2018]

A step beyond: global properties and explainable AI



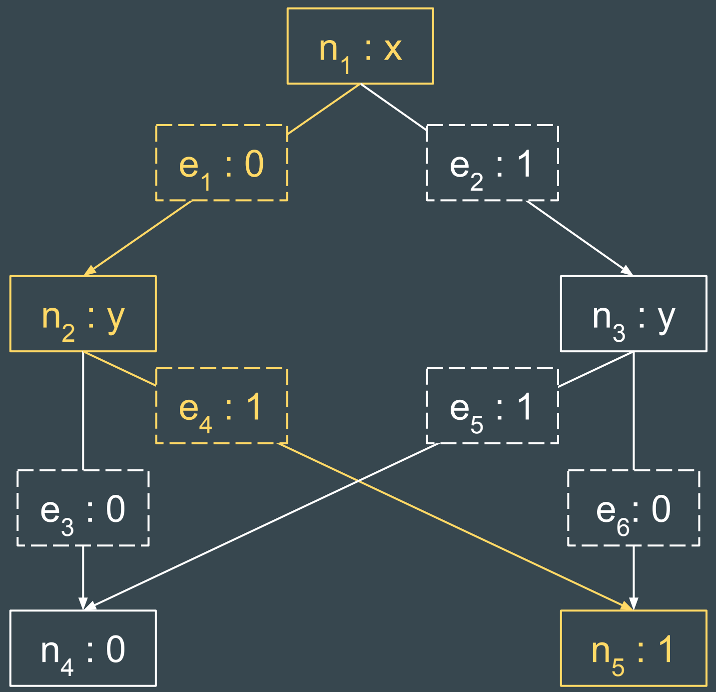
A step beyond: global properties and explainable AI



What kind of queries should we answer?

Is there any instance that is classified positively?

Is there any instance that is classified negatively?

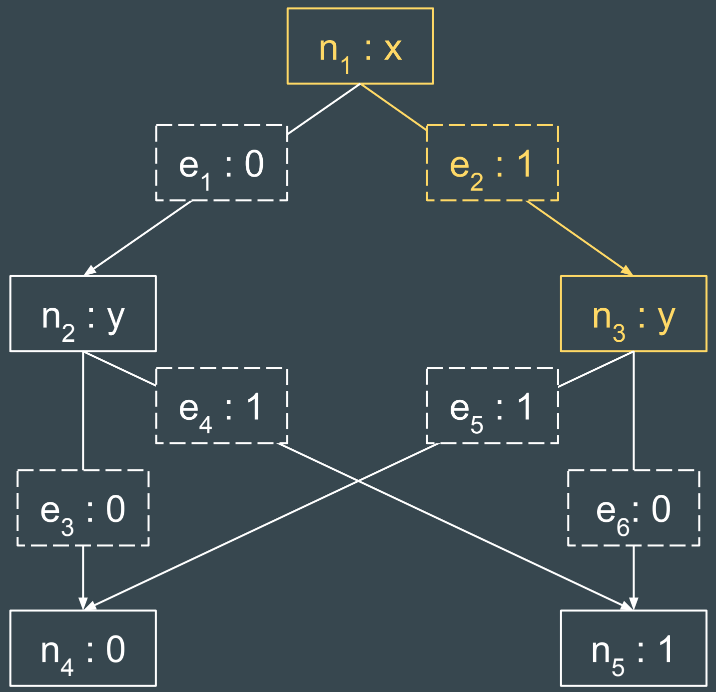


What kind of queries should we answer?

Is there a completion of x ↦ 1 that is classified positively?

Are all the completion of x ↦ 1 classified positively?

- So that x ↦ 1 is a sufficient reason for the positive value

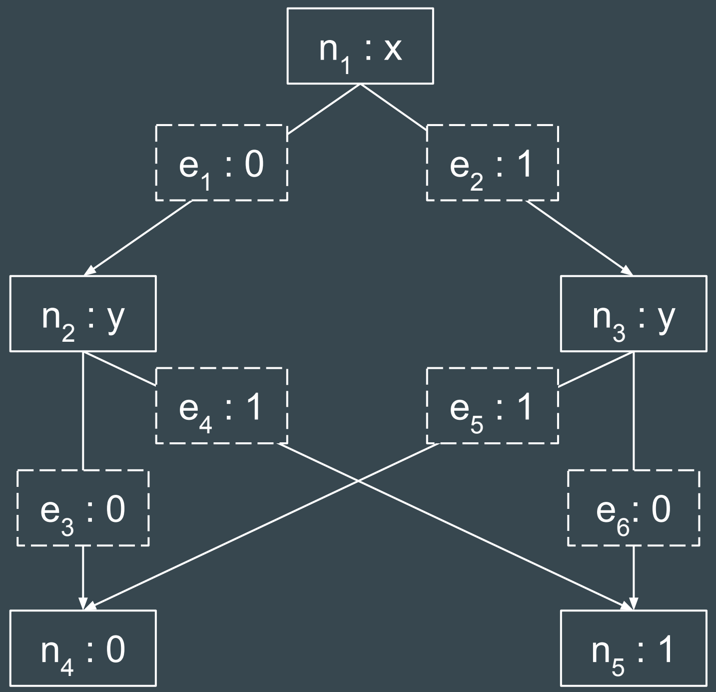


A declarative language for model interpretability

Given an instance classified positively, what is a sufficient reason for it?

What is a minimal sufficient reason for this instance?

Is the model biased with respect to a protected feature?



Some questions to think about

How can a declarative language for model interpretability be defined?

Can such a language be based on *path expressions*? How can such expressions be combined with *quantifiers*?

Can such a language be evaluated efficiently?

- How does this evaluation depend on the structure of the graph? Models can be decision trees, OBDDs, FBDDs, ...

We have gone through technical challenges.
Is that all?

...

Topics we did not cover

At least we are aware of:

- Human visualization of graphs (Upcoming Dagstuhl-Seminar 22031)
- HCI aspects of graph query languages
- Storing and infrastructure issues
- Enterprise and organizational issues
- Governance issues
- Ethical issues

Message: querying is not only a formal / technical topic

Important reminder: humans are always in the loop

Context	Trend	Organizational Need	Technology	Role
Web + Moore's Law	Big Data	Harness and collect data	Commodity distributed computing platforms (e.g. Hadoop)	**Data Engineer**
Big Data + GPU Compute	AI Revolution	Draw value from data	Commodity machine learning (e.g., TensorFlow, SciPy)	**Data Scientist**
AI Revolution + Cloud Computing	Data-Driven Organization, Digital Transformation	Rely on data	Clean, meaningful, data technologies (e.g. knowledge graphs, data wrangling systems, data catalog platforms)	**?**

Important reminder: humans are always in the loop

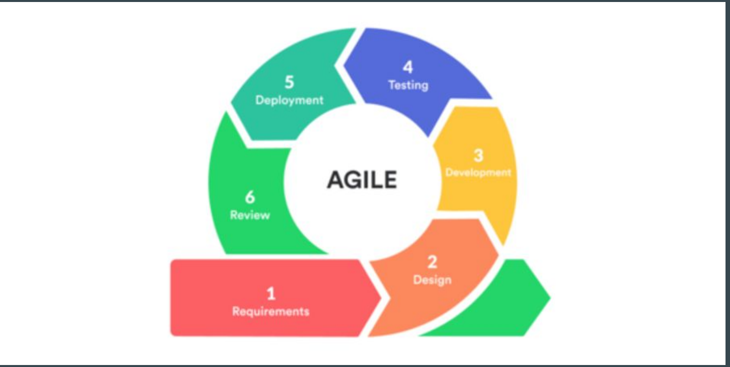
Context	Trend	Organizational Need	Technology	Role
Web + Moore's Law	Big Data	Harness and collect data	Commodity distributed computing platforms (e.g. Hadoop)	**Data Engineer**
Big Data + GPU Compute	AI Revolution	Draw value from data	Commodity machine learning (e.g., TensorFlow, SciPy)	**Data Scientist**
AI Revolution + Cloud Computing	Data-Driven Organization, Digital Transformation	Rely on data	Clean, meaningful, data technologies (e.g. knowledge graphs, data wrangling systems, data catalog platforms)	**Knowledge Scientist**

https://www.knowledgescientist.org/

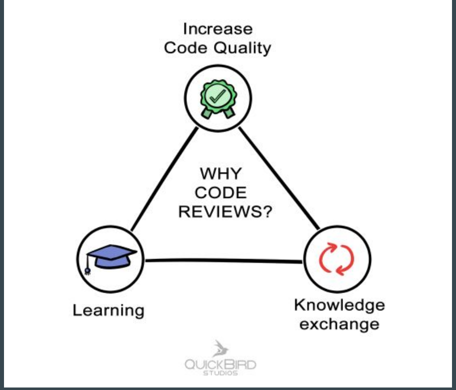
Data is a Team Sport

The most important data isn't data

Agile Data Development



Data Review



Knowledge Report

Knowledge Capture

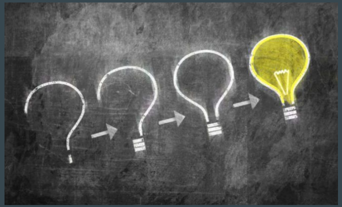
1. Analyze as-is process
2. Collect Documentation
3. Develop Knowledge Report

Business Question

Knowledge Implementation

4. Create/Extend Knowledge Graph Schema
5. Implement Mapping
6. Generate Data Products
7. Validate Data

Enterprise Knowledge Graph



Business Answer

Knowledge Access

8. Build Report
9. Answer Business Question
10. Move to Production

[Sequeda et al. 2019]

References

[Angles and Gutierrez 2008] R. Angles and C. Gutiérrez. *Survey of graph database models*. ACM Comput. Surv. 40(1), 2008.

[Angles et al. 2018] R. Angles, M. Arenas, P. Barceló, P. A. Boncz, G. H. L. Fletcher, C. Gutiérrez, T. Lindaaker, M. Paradies, S. Plantikow, J. F. Sequeda, O. van Rest, and H. Voigt. *G-CORE: A Core for Future Graph Query Languages*. SIGMOD 2018.

[Arenas et at. 2019] M. Arenas, L. A. Croquevielle, R. Jayaram, and C. Riveros. *Efficient Logspace Classes for Enumeration, Counting, and Uniform Generation*. PODS 2019.

[Barceló et al. 2020] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. L. Reutter, and J. P. Silva. *The Logical Expressiveness of Graph Neural Networks*. ICLR 2020.

[Cai et at. 1992] J-Y. Cai, M. Furer, and N. Immerman. *An optimal lower bound on the number of variables for graph identification*. Combinatorica 12(4), 1992.

References

[Chung 2010] F. Chung. *Graph theory in the information age*. Notices of the AMS 57(6), 2010.

[Consens and Mendelzon 1990] M. P. Consens and A. O. Mendelzon. *GraphLog: a Visual Formalism for Real Life Recursion*. PODS 1990.

[Gutierrez and Sequeda 2021] C. Gutierrez and J. Sequeda. *Knowledge Graphs*. Commun. ACM 64(3): 96-104 (2021)

[Morris et al. 2019] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. *Weisfeiler and Leman go neural: higher-order graph neural networks*. AAAI 2019.

[Sequeda et al. 2019] J. F. Sequeda, W. J. Briggs, D. P. Miranker, and W. P. Heideman. *A Pay-as-you-go Methodology to Design and Build Enterprise Knowledge Graphs from Relational Databases*. ISWC 2019.

[Xu et al. 2019] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. *How Powerful are graph neural networks?* ICLR 2019.

Takeaways

- Graphs are not just another data model
  - They have always been here
  - They are not going away
  - This is the right time. We are lucky to be here!
- Knowledge Graphs are more than just graph databases

- Exciting to see results from different areas getting connected
  - Connection of Graph Neural Networks and Graph Query Languages
- Opportunities
  - Explainable AI and the search of a declarative language for interpretability
- Computing is approaching the fine line separating technology from humans. We should be open to learning from other disciplines.