

# Semiparametric Reasoning over Structured Data

Rajarshi (Raj) Das

April 17, 2024



# Who am I? 🙌

- ◆ Currently a researcher at Amazon
- ◆ PhD at UMass Amherst with Prof. Andrew McCallum
- ◆ Postdoc at University of Washington with Prof. Hanna Hajishirzi
- ◆ Thesis on “Semiparametric Contextual Reasoning Models for QA over KBs and Text”



# Structured Databases

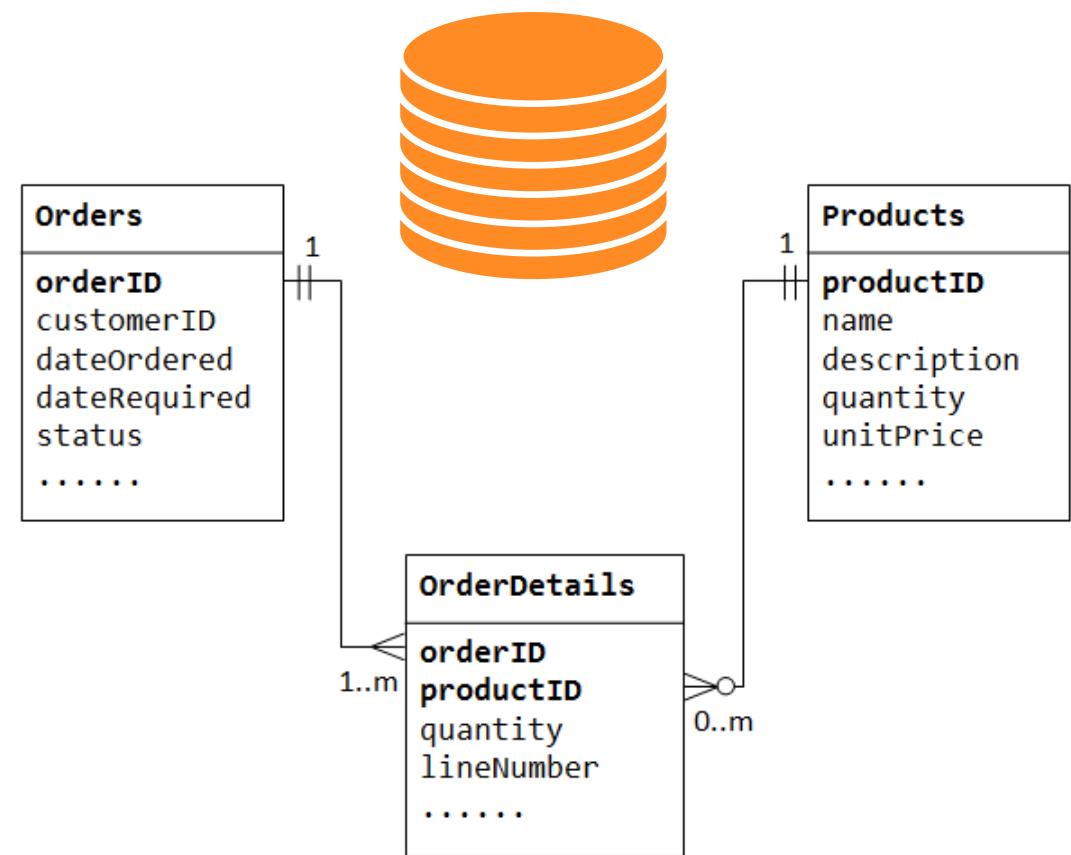
# Structured Databases

- ◆ Structured data is ubiquitous.

# Structured Databases

- ◆ Structured data is ubiquitous.

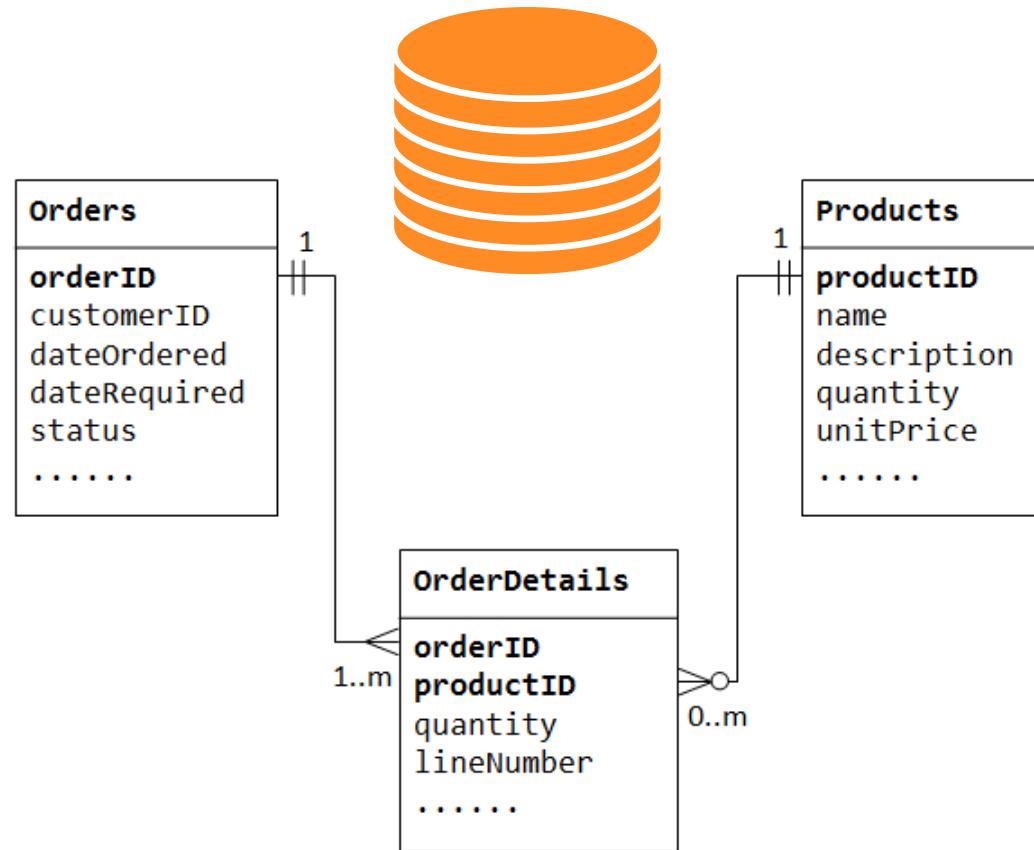
Relational Databases



# Structured Databases

- ◆ Structured data is ubiquitous.

Relational Databases



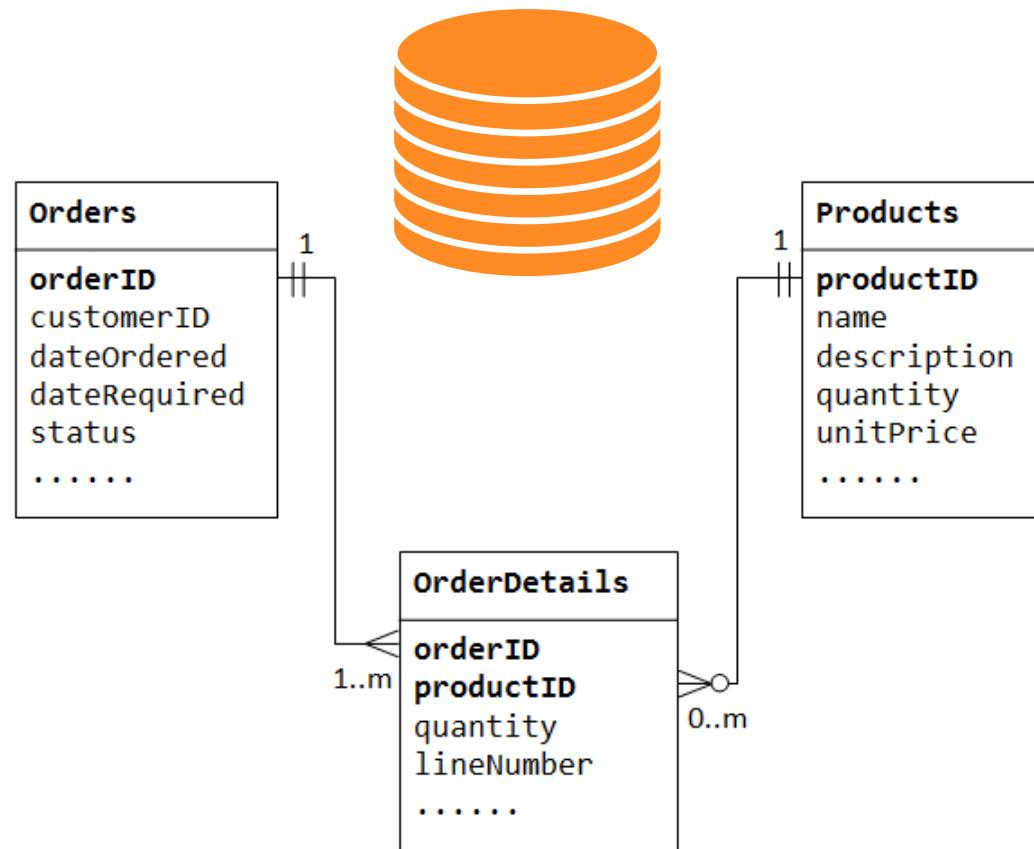
Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit

# Structured Databases

- ◆ Structured data is ubiquitous.

Relational Databases



Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit

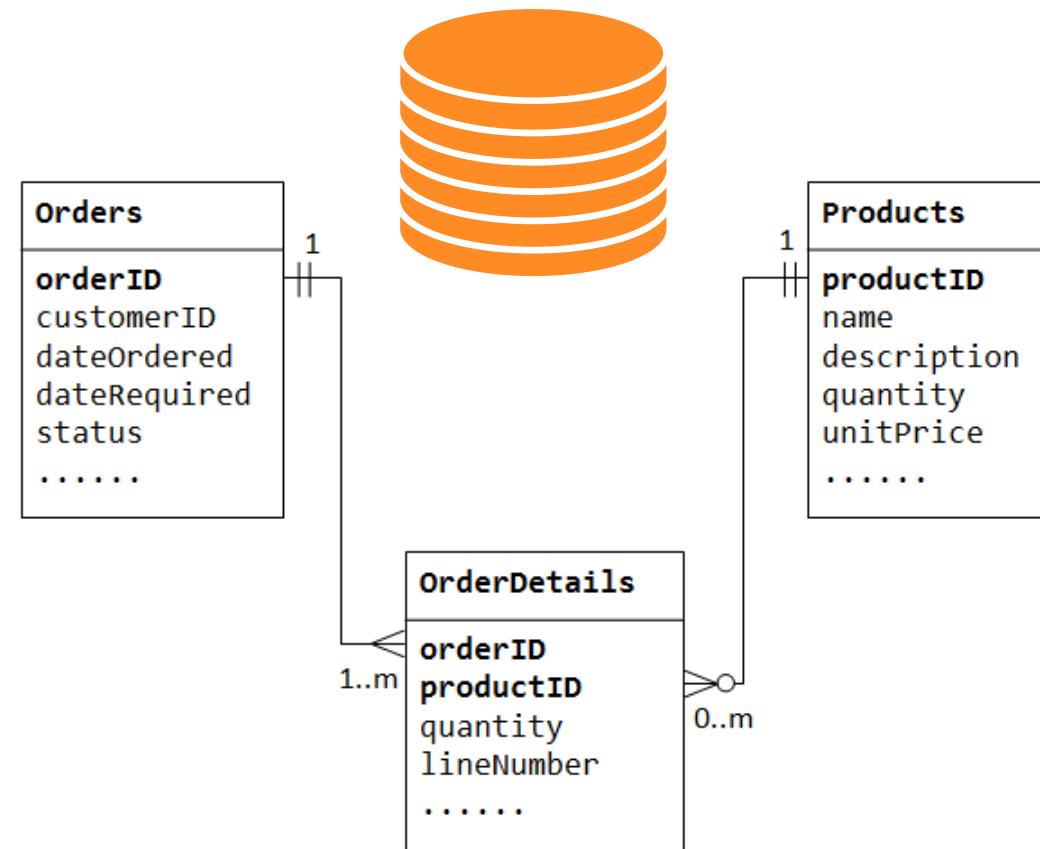


{ json }

# Structured Databases

- ◆ Structured data is ubiquitous.

Relational Databases



Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit

Graphs

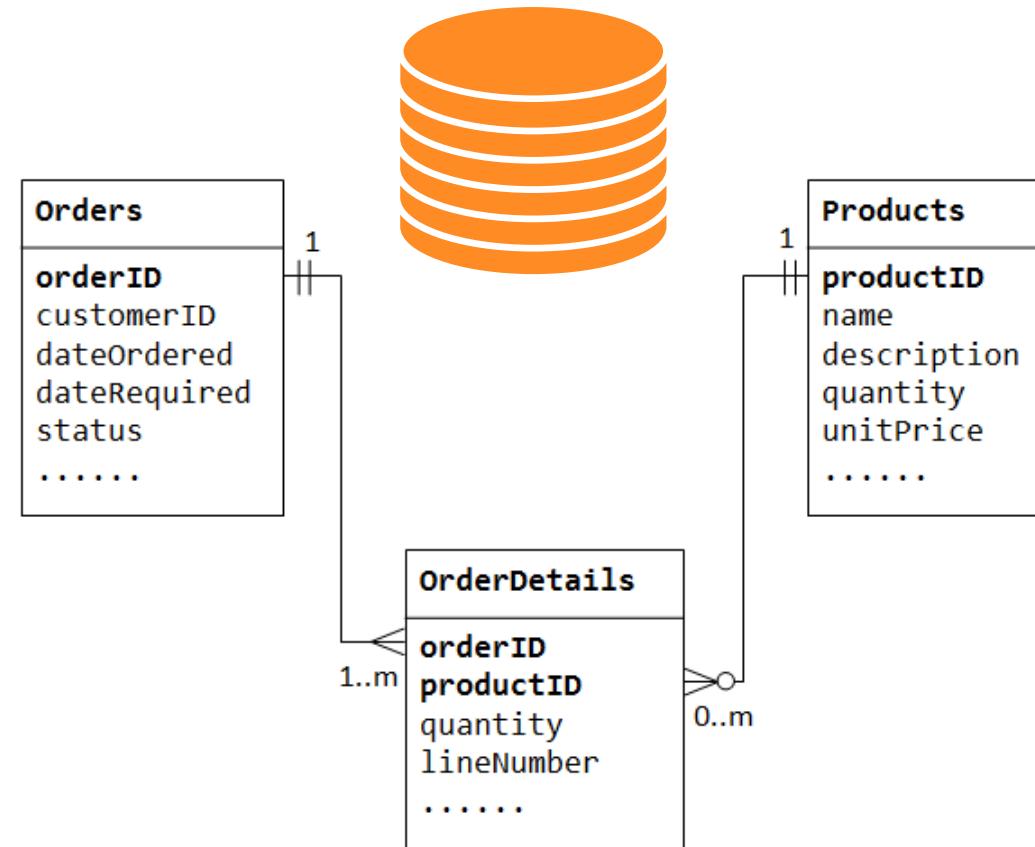


{ json }

# Structured Databases

♦ Structured data is ubiquitous.

Relational Databases



Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit



Graphs

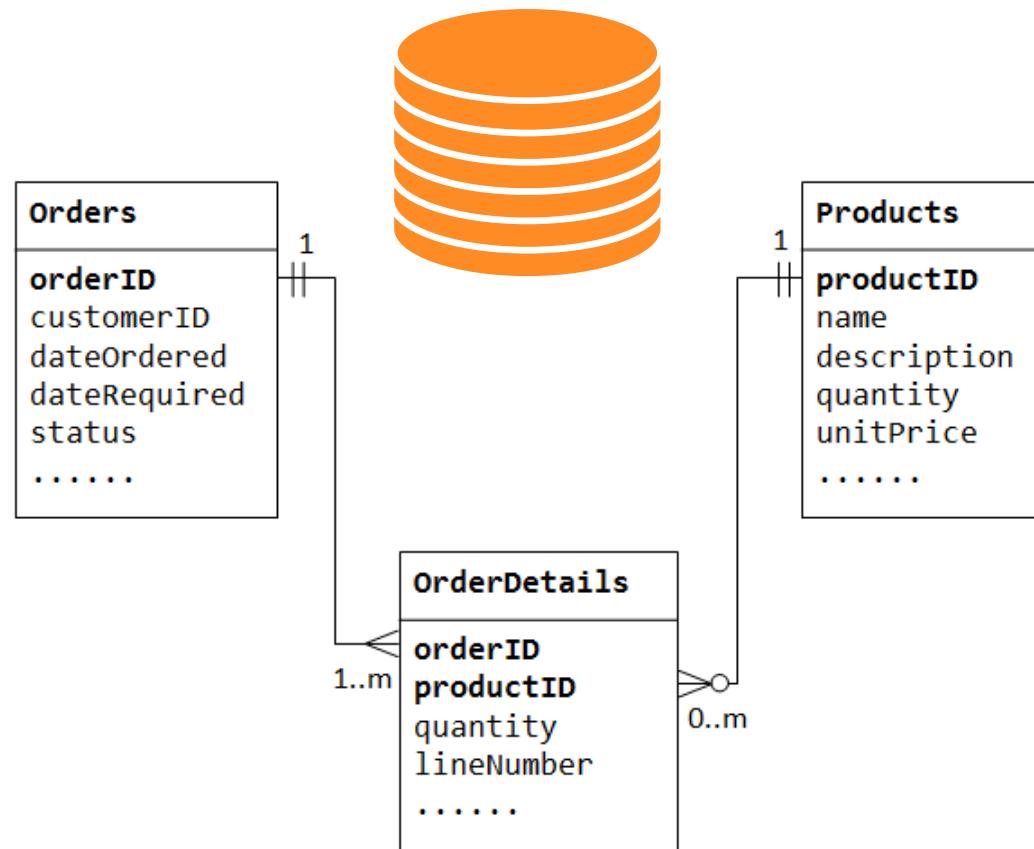


{ json }

# Structured Databases

◆ Structured data is ubiquitous.

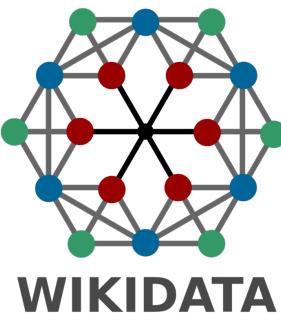
Relational Databases



Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit

Freebase™



Graphs

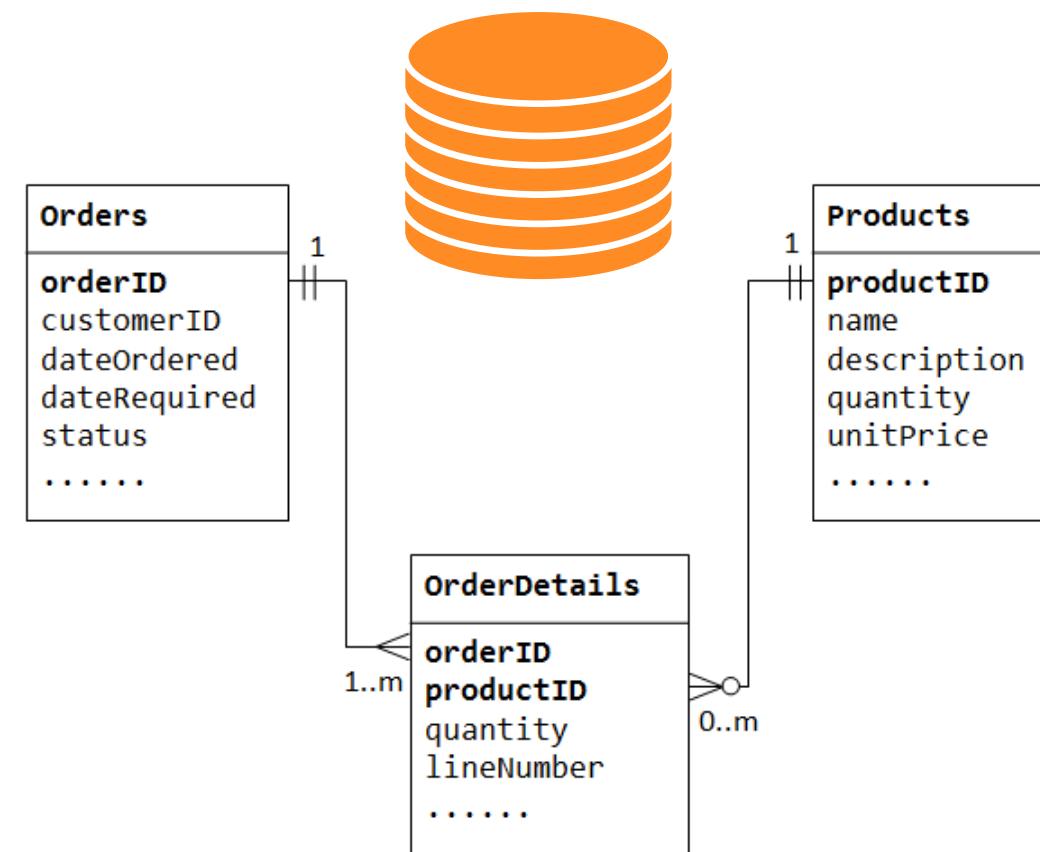


{ json }

# Structured Databases

◆ Structured data is ubiquitous.

Relational Databases



Web Tables

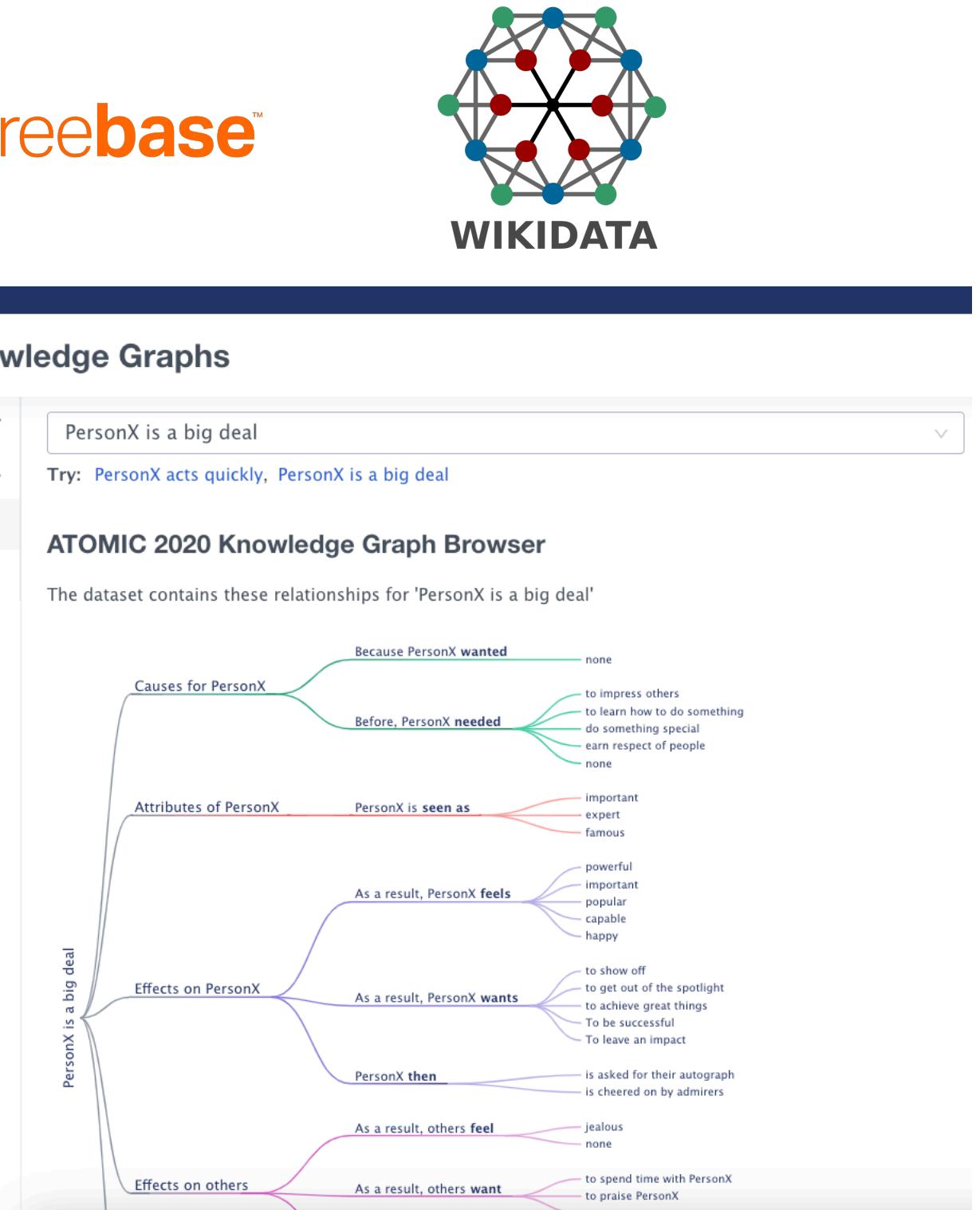
Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit



{ json }

Freebase

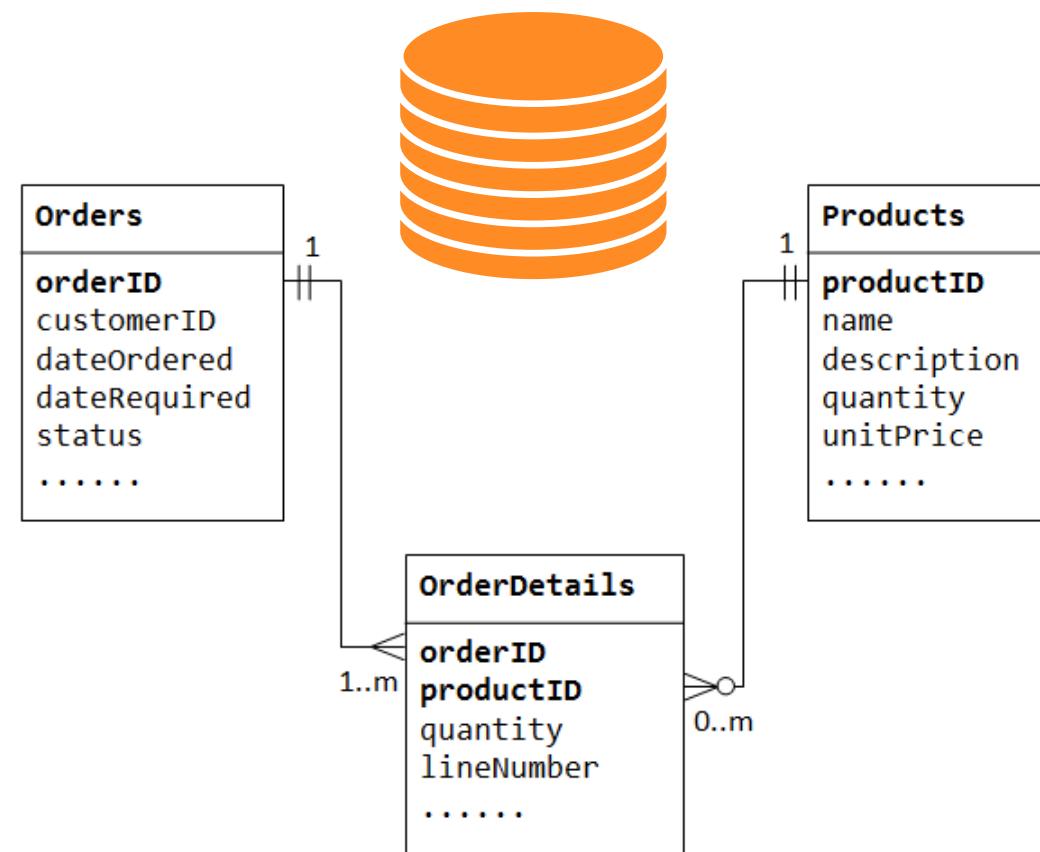
Graphs



# Structured Databases

♦ Structured data is ubiquitous.

Relational Databases



Web Tables

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
p5.48xlarge	\$98.32	192	2048 GiB	8 x 3840 GB SSD	3200 Gigabit
p4d.24xlarge	\$32.7726	96	1152 GiB	8 x 1000 SSD	400 Gigabit
p3.2xlarge	\$3.06	8	61 GiB	EBS Only	Up to 10 Gigabit
p3.8xlarge	\$12.24	32	244 GiB	EBS Only	10 Gigabit
p3.16xlarge	\$24.48	64	488 GiB	EBS Only	25 Gigabit
p2.xlarge	\$0.90	4	61 GiB	EBS Only	High
p2.8xlarge	\$7.20	32	488 GiB	EBS Only	10 Gigabit
p2.16xlarge	\$14.40	64	732 GiB	EBS Only	20 Gigabit
g6.xlarge	\$0.8048	4	16 GiB	1 x 250 GB NVMe SSD	Up to 10 Gigabit
g6.2xlarge	\$0.9776	8	32 GiB	1 x 450 GB NVMe SSD	Up to 10 Gigabit
g6.4xlarge	\$1.3232	16	64 GiB	1 x 600 GB NVMe SSD	Up to 25 Gigabit
g6.8xlarge	\$2.0144	32	128 GiB	2 x 450 GB NVMe SSD	25 Gigabit



{ json }



3

Freebase™

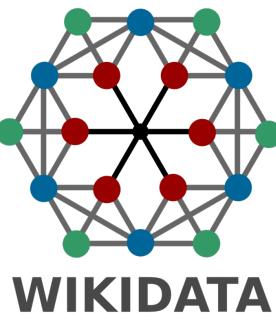
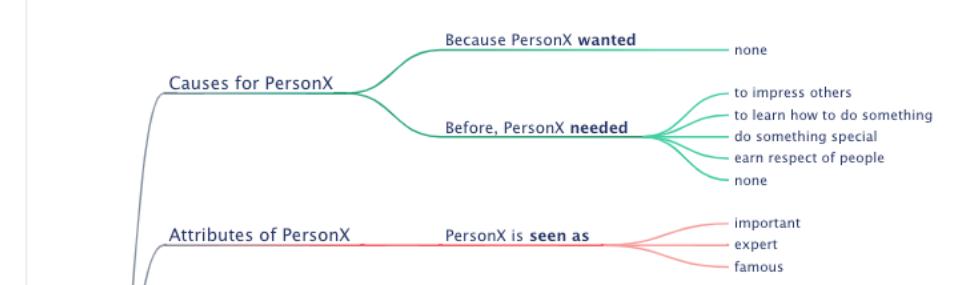
AI2 Allen Institute for AI  
Mosaic Knowledge Graphs

Model Knowledge Graph ATOMIC 2020 About

PersonX is a big deal Try: PersonX acts quickly, PersonX is a big deal

ATOMIC 2020 Knowledge Graph Browser

The dataset contains these relationships for 'PersonX is a big deal'



Graphs



KNOWLEDGE GRAPH

# Knowledge Graphs

## Irène Joliot-Curie

From Wikipedia, the free encyclopedia

Irène Joliot-Curie (French: [iʁɛn zɔljɔ kɥʁi]; 12 September 1897 – 17 March 1956) was a French chemist, physicist, and a politician of partly Polish ancestry, the elder daughter of Marie Curie and Pierre Curie, and the wife of Frédéric Joliot-Curie. She won the Nobel Prize in Chemistry in 1935 for their discovery of artificial radioactivity. This made the Curies the family with the most Nobel Prizes.

Pierre, are also among the commissioners of the new French Alternative Energies and Atomic Energy Commission (CEA), created by de Gaulle and the first members of the French Republic. She died on 17 March 1956 from an acute leukemia linked to her exposure to radioactive materials.

**Contents [hide]**

- 1 Biography
  - 1.1 Early life and education
  - 1.2 World War I
  - 1.3 Research
  - 1.4 Political views
  - 1.5 Personal life
  - 1.6 Notable honours
- 2 See also
- 3 References
- 4 Further reading
- 5 External links

**Nobel Prize in Chemistry**

**1903** ← **Won** → **1911**

**Radioactivity**

**Discov -ered**

**Research**

**Spouse**

**Father**

**Mother**

**Research**

**Won**

**Won**

**Won**

**Paris**

**Located\_In**

**Paris**

**Irene Joliot-Curie**

**Frederic Joliot-Curie**

**1935**

**Univ of Paris**

**Nobel Prize in Chemistry**

**Marie Curie**

**Pierre Curie**

**Nuclear Physics**

**Radioactivity**

**Nobel Prize in Physics**

## Marie Curie

From Wikipedia, the free encyclopedia

This article is about the Polish-French physicist. For other uses, see Marie Curie (disambiguation).

Marie Skłodowska Curie (French: [maʁi kʁyɛ]; Polish: [maˈrjɛ skwɔˈdɔfska]; 7 November 1867 – 4 July 1934), was a Polish and naturalized-French physicist and chemist, known for her contributions to the study of radioactivity. As the first of the Curie family to win a Nobel Prize, she was the first woman to win a Nobel Prize in two different fields. She was the first woman to become a professor at the University of Paris in 1906.<sup>[4]</sup>

She was born in Warsaw, in what was then the Kingdom of Poland, part of the Russian Empire. She studied at Warsaw's Jagiellonian University and began her practical scientific training in Warsaw. In 1891, aged 24, she followed her elder sister Bronisława to study in Paris, where she earned two further degrees and conducted her subsequent scientific work.

In 1895 she married the French physicist Pierre Curie, and she shared the 1903 Nobel Prize in Physics with him and with the physicist Henri Becquerel for their pioneering work developing the theory of "radioactivity"—a term coined.<sup>[5][6]</sup> In 1906 Pierre Curie died in a Paris street accident. Marie won the 1911 Nobel Prize in Chemistry for her discovery of the elements polonium and radium, using techniques she invented for isolating radioactive isotopes.

During World War I, she developed mobile radiography units to provide X-ray services to military hospitals. She taught X-ray techniques to medical students and took them on visits to Poland.<sup>[8]</sup> She named the first chemical element she discovered polonium, after her native country.<sup>[9]</sup>

Marie Curie died in 1934, aged 66, at the Sancellemoz sanatorium in Passy (Haute-Savoie), France, of aplastic anemia from exposure to radiation in the course of her scientific research and in the course of her radiological work at field hospitals during World War I.<sup>[10]</sup> In addition to her Nobel Prizes, she has received numerous other honours and tributes; in 1932 she became the first woman to be entombed on her own merits in Paris' Panthéon.<sup>[11]</sup> Poland declared 2011 as the Year of Marie Curie during the International Year of Chemistry. She is the subject of numerous biographical works, where she is also known as Madame Curie.

## Paris

From Wikipedia, the free encyclopedia

This article is about the capital of France. For other uses, see Paris (disambiguation).

Paris (French pronunciation: [paul] (listen)) is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more than 105 square kilometres (41 square miles).<sup>[4]</sup> Since the 17th century, Paris has been one of Europe's major centres of finance, diplomacy, commerce, fashion, gastronomy, science and arts. The City of Paris is the centre and seat of government of the Île-de-France, or Paris Region, which has an estimated population of 12,174,880, or about 18 percent of the population of France as of 2017.<sup>[5]</sup> The Paris Region had a GDP of €709 billion (\$808 billion) in 2017.<sup>[6]</sup> According to the Economist Intelligence Unit Worldwide Cost of Living Survey in 2018, Paris was the second most expensive city in the world, after Singapore and ahead of Zürich, Hong Kong, Oslo and Geneva.<sup>[7]</sup> Another source ranked Paris as most expensive, on a par with Singapore and Hong Kong, in 2018.<sup>[8][9]</sup>

Paris is a major railway, highway and air-transport hub served by two international airports: Paris–Charles de Gaulle (the second busiest airport in Europe) and Paris–Orly.<sup>[10][11]</sup> Opened in 1900, the city's subway system, the Paris Métro, serves 5.23 million passengers daily,<sup>[12]</sup> it is the second busiest metro system in Europe after the Moscow Metro. Gare du Nord is the 24th busiest railway station in the world, but the first located outside

## Radioactive decay

From Wikipedia, the free encyclopedia

For particle decay in a more general context, see Particle decay. For more information on hazards of various kinds of radioactive decay, see Radioactive hazard. "Radioactive" and "Radioactivity" redirect here. For other uses, see Radioactive (disambiguation) and Radioactivity (disambiguation).

Radioactive decay (also known as nuclear decay, radioactivity, radioactive disintegration or nuclear disintegration) is the process by which an unstable nucleus is considered radioactive. Three of the most common types of decay are alpha decay, beta decay, and gamma decay.

Radioactive decay is a stochastic (i.e. random) process at the level of single atoms. According to quantum theory, it is impossible to predict exactly when a specific atom will decay, although the rate can be expressed as a decay constant or as half-life.

The decaying nucleus emits particles and energy in the form of radiation. The decaying nucleus is called a parent nucleus, and it transforms into a different nucleus, called the daughter nucleus. This process is called decay. There are three main types of decay:

- Alpha decay
- Beta decay
- Gamma decay

- (i) beta-minus decay
- (ii) beta-plus decay
- In gamma decay

Awarded for Contributions that have conferred the greatest benefit to humankind in the areas of Physics, Chemistry, Physiology or Medicine, Literature, and Peace.

Country Sweden (all prizes except the Peace Prize)

Presented by Norway (Peace Prize only)

Royal Swedish Academy of Sciences (Physics and Chemistry)

Models of the nucleus

Nucleus • Nucleons (p, n) • Nuclear matter • Nuclear force • Nuclear structure • Nuclear reaction

Coordinates: 48°51'24"N 2°21'08"E

Paris Capital city, department and commune

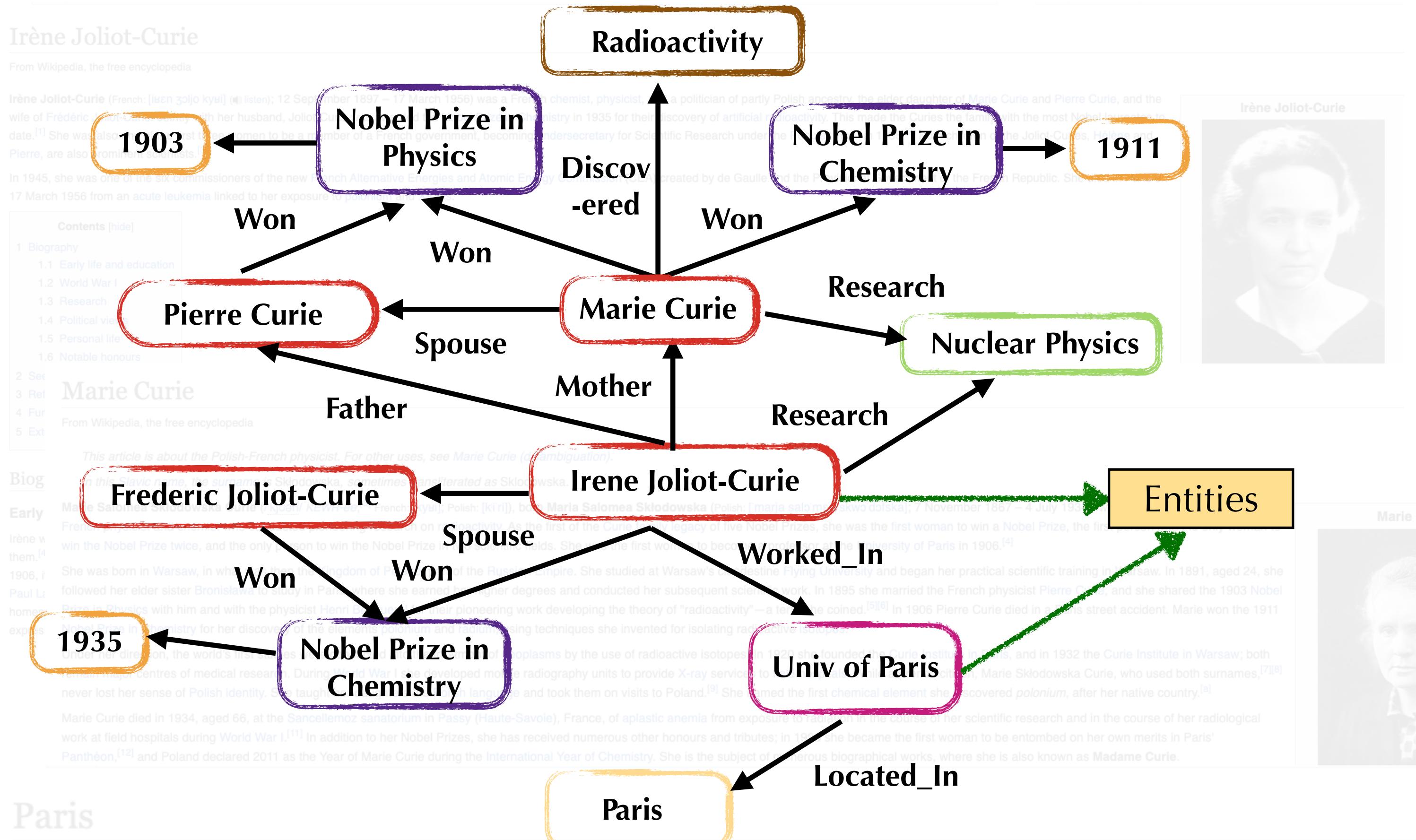
Paris

Paris

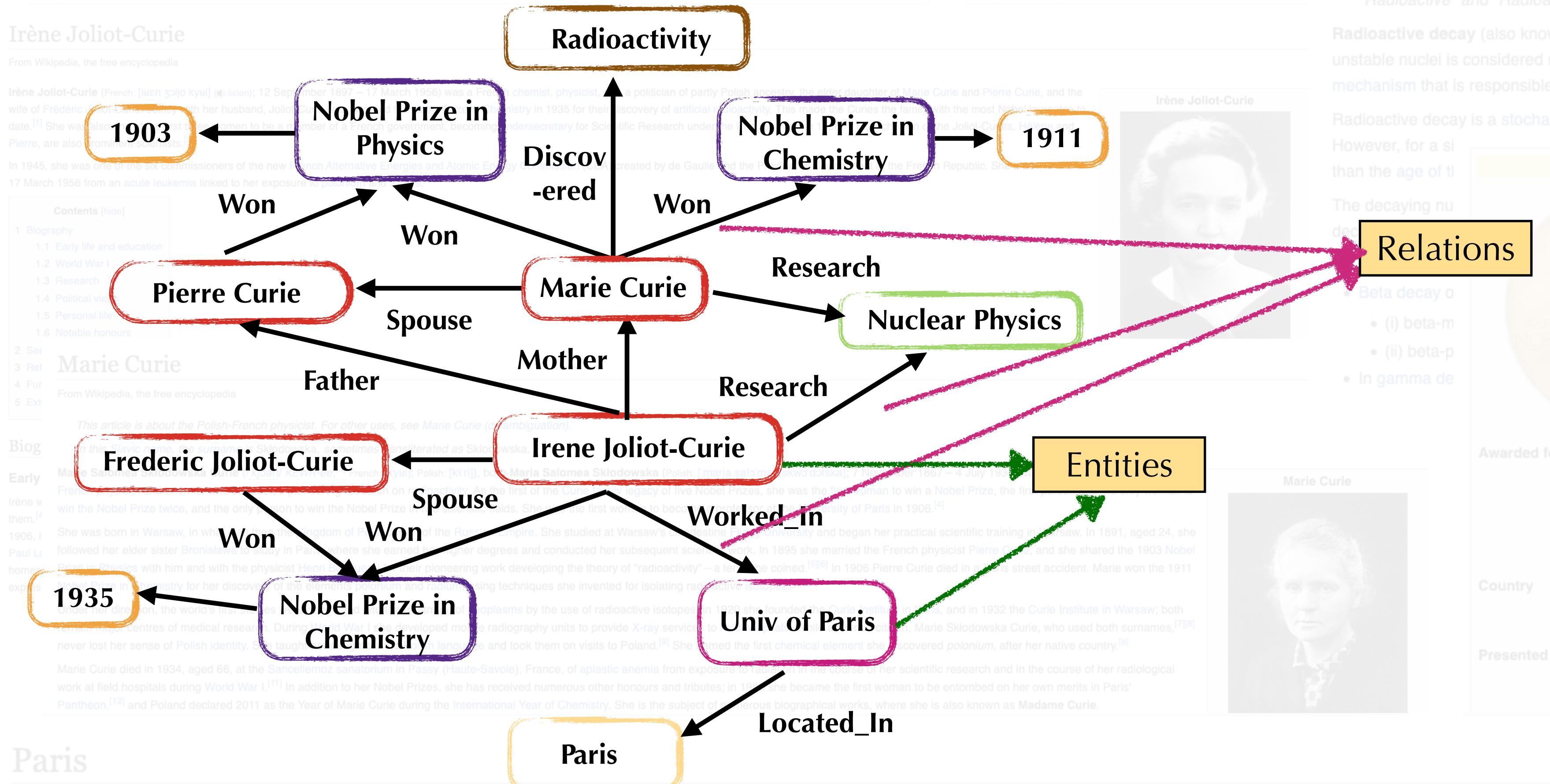
Paris

Paris

# Knowledge Graphs



# Knowledge Graphs



The decaying nucleus emits an alpha particle, which is an helium nucleus. This process is called alpha decay. The resulting nucleus is called an alpha isotope<sup>[note 1]</sup>, and the process produces at least one different number of protons or neutrons (or both). When an alpha particle is emitted, the mass number of the nucleus decreases by four.

# **Knowledge Graphs - Are they Still Relevant?**

# Knowledge Graphs - Are they Still Relevant?

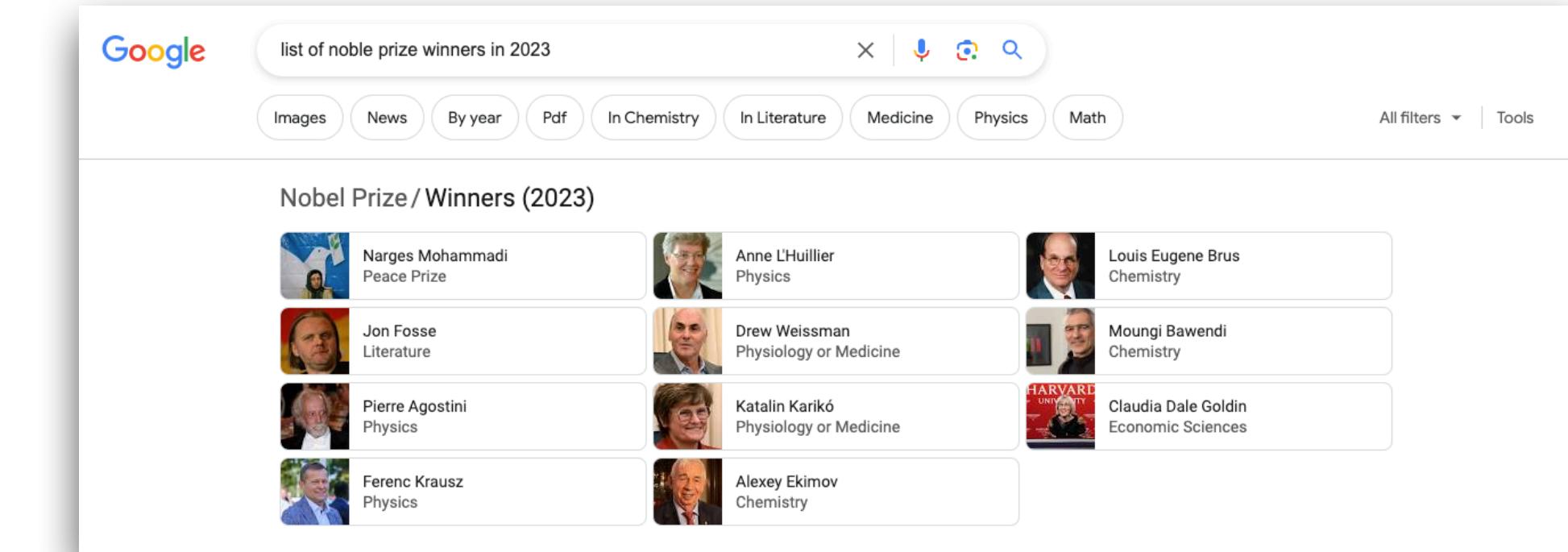
- ♦ Yes!

# Knowledge Graphs - Are they Still Relevant?

- ◆ Yes!
- ◆ Industry:
  - ◆ Question Answering powering chatbots and search engines
  - ◆ Product Knowledge Graphs
  - ◆ Cloud resources

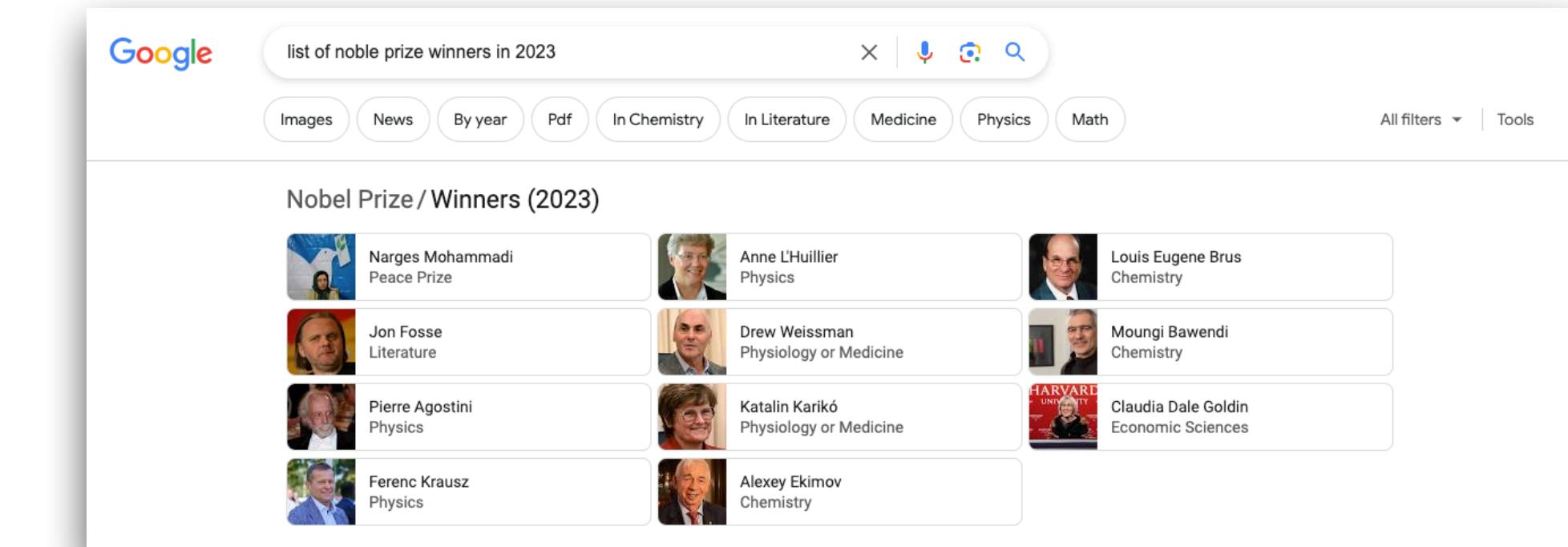
# Knowledge Graphs - Are they Still Relevant?

- ◆ Yes!
- ◆ Industry:
  - ◆ Question Answering powering chatbots and search engines
  - ◆ Product Knowledge Graphs
  - ◆ Cloud resources



# Knowledge Graphs - Are they Still Relevant?

- ◆ Yes!
- ◆ Industry:
  - ◆ Question Answering powering chatbots and search engines
  - ◆ Product Knowledge Graphs
  - ◆ Cloud resources

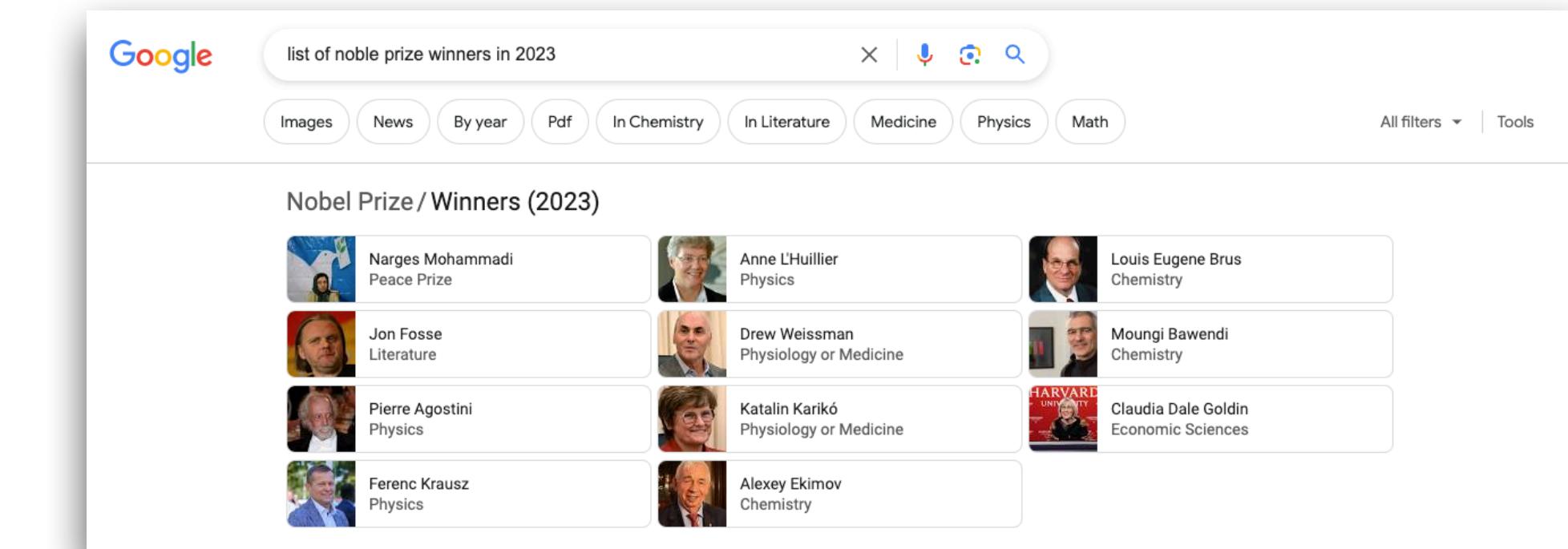


5



# Knowledge Graphs - Are they Still Relevant?

- ◆ Yes!
- ◆ Industry:
  - ◆ Question Answering powering chatbots and search engines
  - ◆ Product Knowledge Graphs
  - ◆ Cloud resources
- ◆ Specialized domains:
  - ◆ Drug Discovery
  - ◆ Material Science



5



# Querying KGs

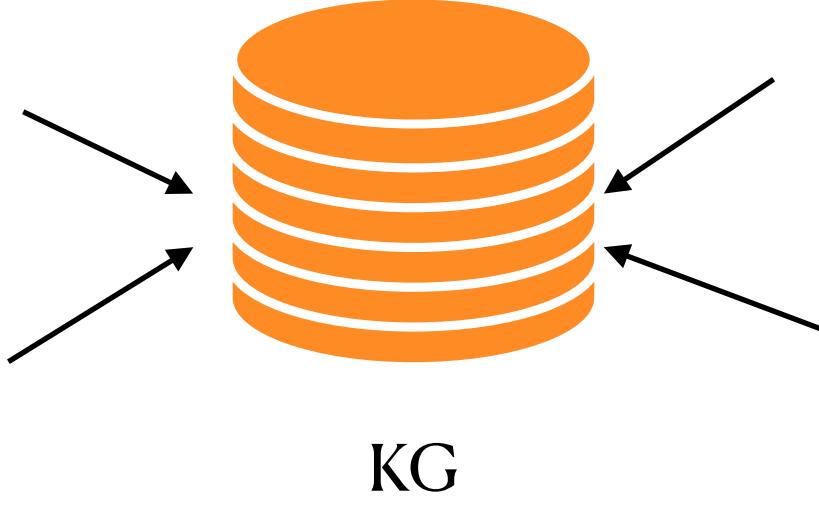
# Querying KGs

- ♦ How do we get information from KGs?

# Querying KGs

- ♦ How do we get information from KGs?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
SELECT DISTINCT ?x0 WHERE {  
?x0 :type.object.type :opera.opera_designer_role .  
?x1 :type.object.type :opera.opera_designer_gig .  
VALUES ?x2 { :m.0pm2fgf }  
?x1 :opera.opera_designer_gig.design_role ?x0 .  
?x2 :opera.opera_production.designers ?x1 .  
FILTER ( ?x0 != ?x1 && ?x0 != ?x2 && ?x1 != ?x2 ) } }
```



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
SELECT DISTINCT ?x0 WHERE {  
?x0 :type.object.type :measurement_unit.measurement_system .  
VALUES ?x1 { :m.02sj5gj }  
?x0 :measurement_unit.measurement_system.radiance_units ?x1 .  
FILTER ( ?x0 != ?x1 ) }}
```

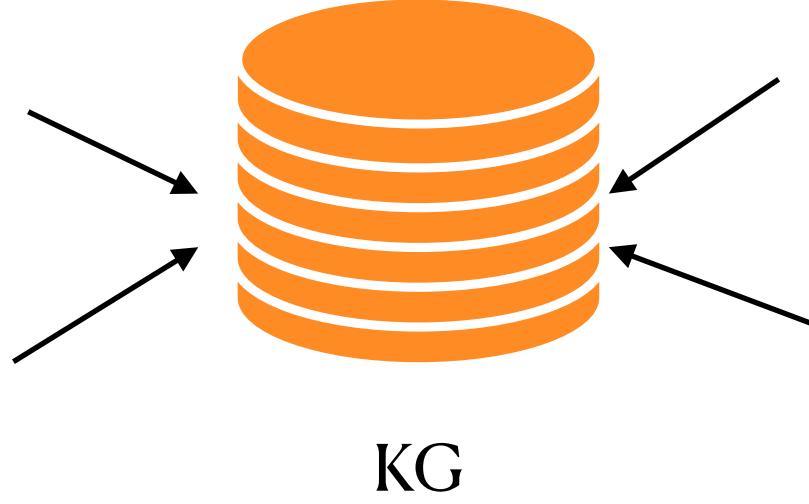
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE { \nSELECT DISTINCT ?x0  
WHERE {  
?x0 :type.object.type :rail.rail_network .  
VALUES ?x1 { :m.03qcvdj }  
?x0 :rail.rail_network.railways ?x1 .  
FILTER ( ?x0 != ?x1 ) \n}\n}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (COUNT(?x0) AS ?value) WHERE { \nSELECT DISTINCT ?  
x0 WHERE {  
?x0 :type.object.type :cvg.computer_videogame . \nVALUES  
?x1 { :m.02hptvh }  
?x1 :cvg.cvg_designer.games_designed ?x0 . FILTER ( ?  
x0 != ?x1 ) \n}\n}
```

# Querying KGs

- ♦ How do we get information from KGs?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
SELECT DISTINCT ?x0 WHERE {  
?x0 :type.object.type :opera.opera_designer_role .  
?x1 :type.object.type :opera.opera_designer_gig .  
VALUES ?x2 { :m.0pm2fgf }  
?x1 :opera.opera_designer_gig.design_role ?x0 .  
?x2 :opera.opera_production.designers ?x1 .  
FILTER ( ?x0 != ?x1 && ?x0 != ?x2 && ?x1 != ?x2 )}}
```



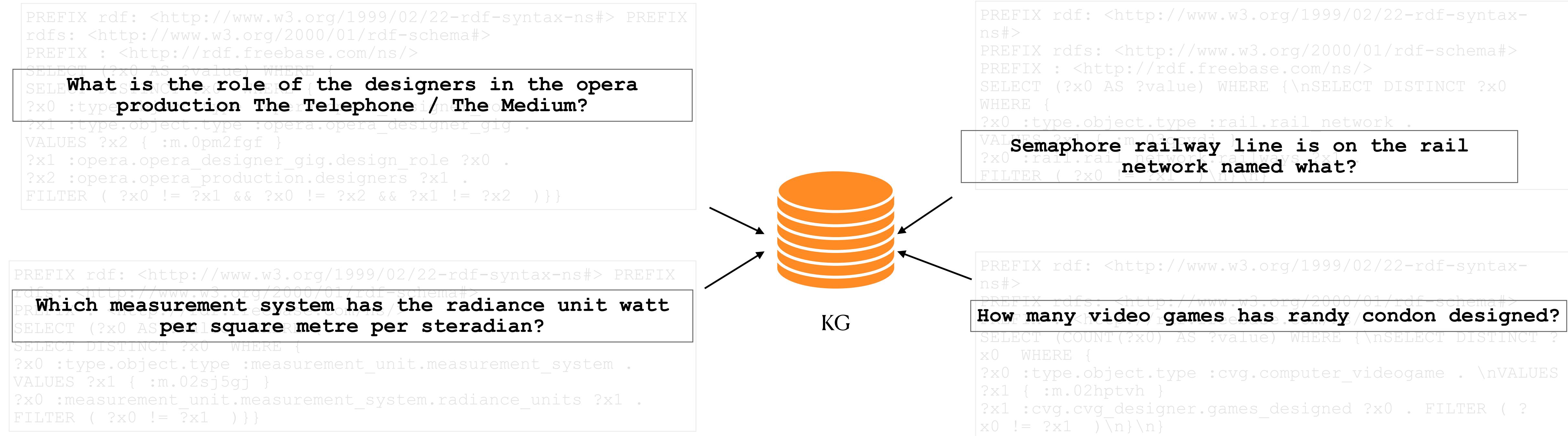
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
SELECT DISTINCT ?x0 WHERE {  
?x0 :type.object.type :measurement_unit.measurement_system .  
VALUES ?x1 { :m.02sj5gj }  
?x0 :measurement_unit.measurement_system.radiance_units ?x1 .  
FILTER ( ?x0 != ?x1 )}}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
SELECT DISTINCT ?x0 WHERE {  
?x0 :type.object.type :rail.rail_network .  
VALUES ?x1 { :m.03qcvdj }  
?x0 :rail.rail_network.railways ?x1 .  
FILTER ( ?x0 != ?x1 )}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-  
ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (COUNT(?x0) AS ?value) WHERE {  
SELECT DISTINCT ?  
x0 WHERE {  
?x0 :type.object.type :cvg.computer_videogame .  
VALUES ?x1 { :m.02hptvh }  
?x1 :cvg.cvg_designer.games_designed ?x0 .  
FILTER ( ?  
x0 != ?x1 )}
```

# Querying KGs

- # ◆ How do we get information from KGs?

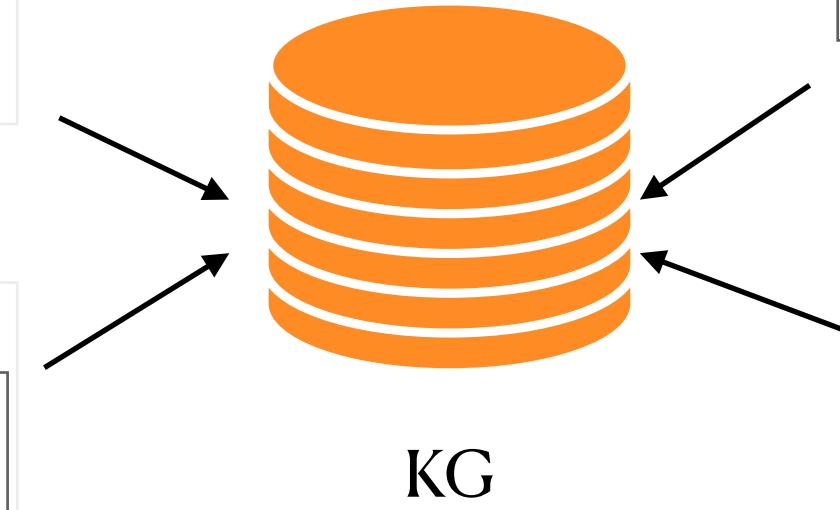


# Querying KGs

- ♦ How do we get information from KGs?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
  SELECT DISTINCT ?x0 WHERE {  
    ?x0 :type.production ?x1 .  
    ?x1 :type.object.type :opera.opera_designer_gig .  
    VALUES ?x2 { :m.0pm2fgf }  
    ?x1 :opera.opera_designer_gig.design_role ?x0 .  
    ?x2 :opera.opera_production.designers ?x1 .  
    FILTER ( ?x0 != ?x1 && ?x0 != ?x2 && ?x1 != ?x2 ) }}  
  
What is the role of the designers in the opera production The Telephone /n The Medium?
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX  
rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
  SELECT DISTINCT ?x0 WHERE {  
    ?x0 :type.object.type :measurement_unit.measurement_system .  
    VALUES ?x1 { :m.02sj5gj }  
    ?x0 :measurement_unit.measurement_system.radiance_units ?x1 .  
  }}  
  
Which measurement system has the radiance unit watt per square metre per steradian?
```



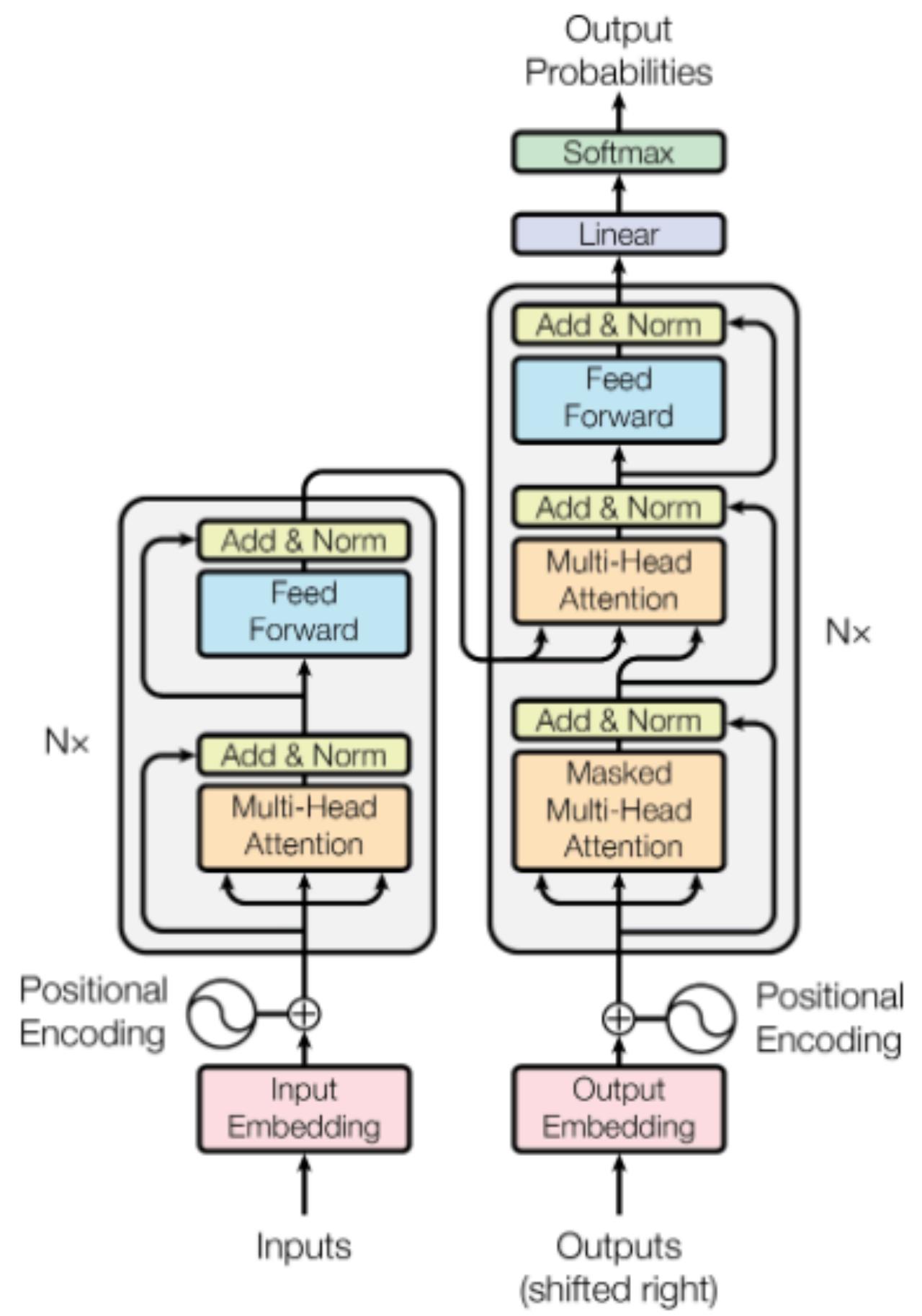
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (?x0 AS ?value) WHERE {  
  SELECT DISTINCT ?x0 WHERE {  
    ?x0 :type.object.type :rail.rail_network .  
    VALUES ?x1 { :m.02hptvh }  
    ?x0 :rail.rail_network.railways ?x1 .  
    FILTER ( ?x0 != ?x1 ) }}  
  
Semaphore railway line is on the rail network named what?
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX : <http://rdf.freebase.com/ns/>  
SELECT (COUNT(?x0) AS ?value) WHERE {  
  SELECT DISTINCT ?x0 WHERE {  
    ?x0 :type.object.type :cvg.computer_videogame .  
    VALUES ?x1 { :m.02hptvh }  
    ?x0 :cvg.computer_videogame.designer ?x1 .  
  }}  
  
How many video games has randy condon designed?
```

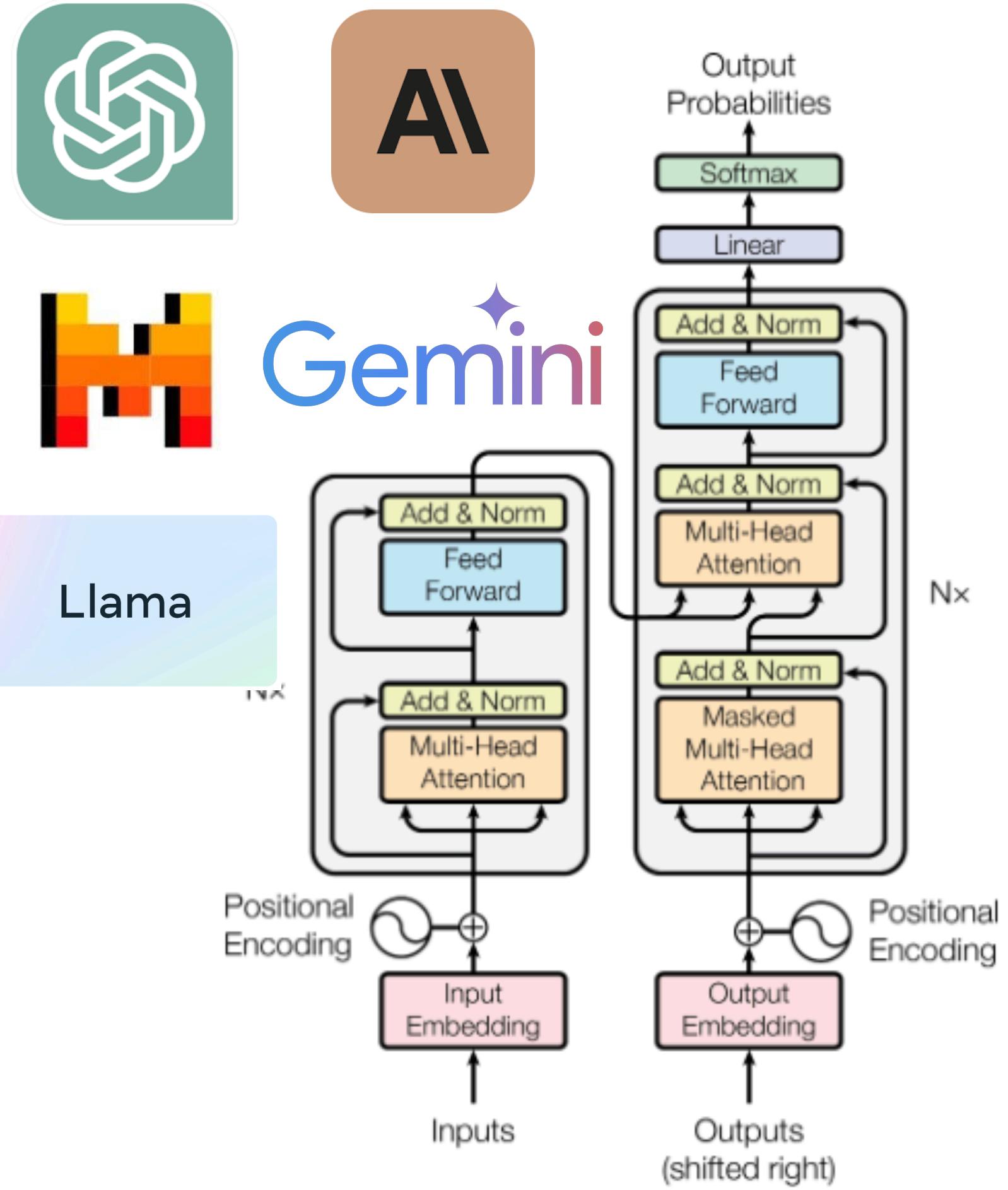
**Goal #1 : Make NL Interfaces for querying KGs**

# Parametric Models for NLP

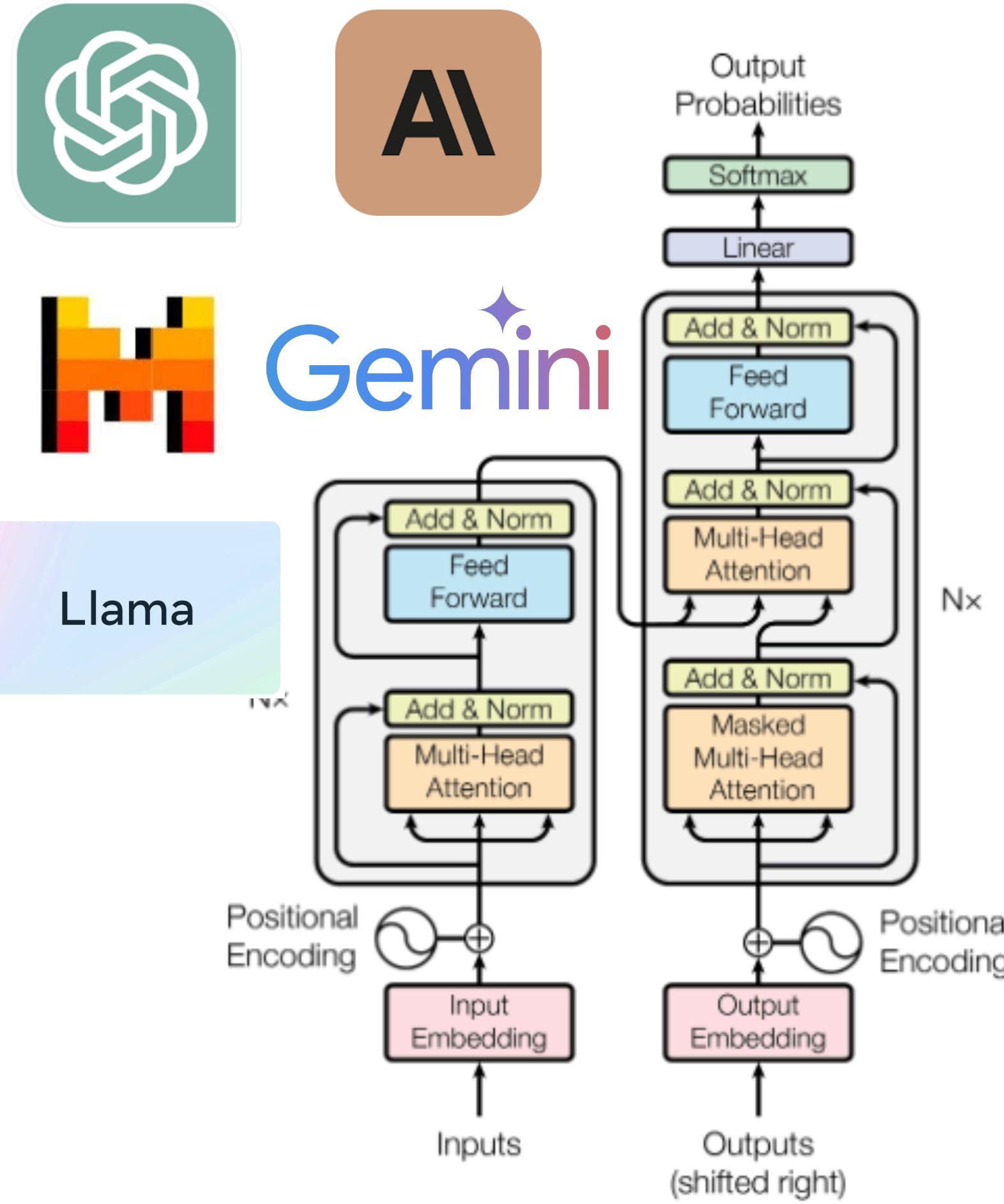
# Parametric Models for NLP



# Parametric Models for NLP

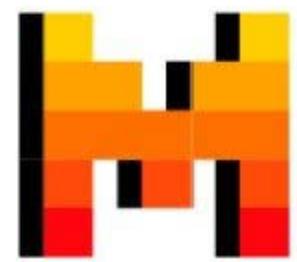


# Parametric Models for NLP



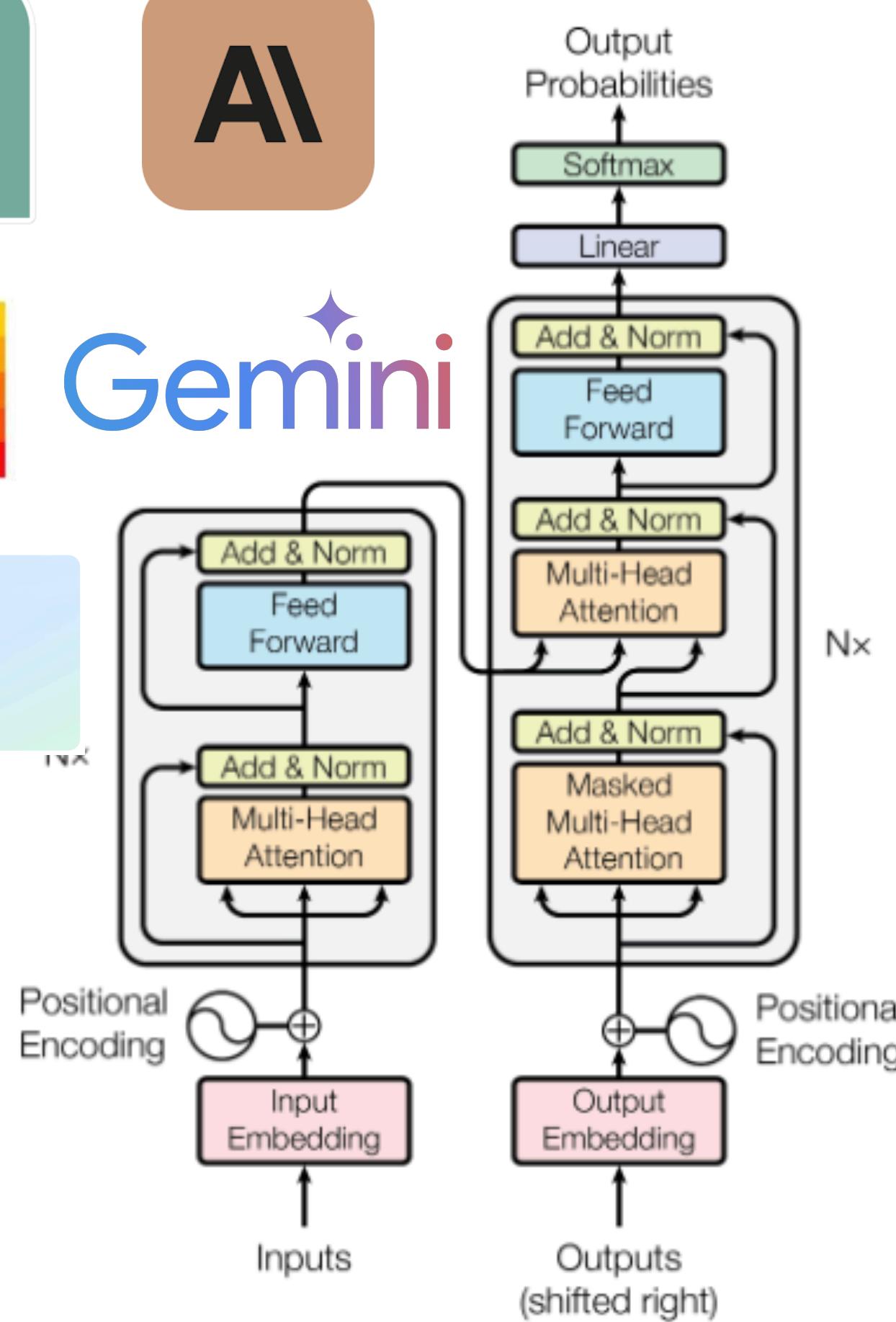
- ✓ Dramatic success of pre-trained LMs on many NLP tasks
- ✓ Model parameters pack a lot of knowledge
- ✓ Emergent abilities at massive scales.

# Parametric Models for NLP



Gemini

Llama

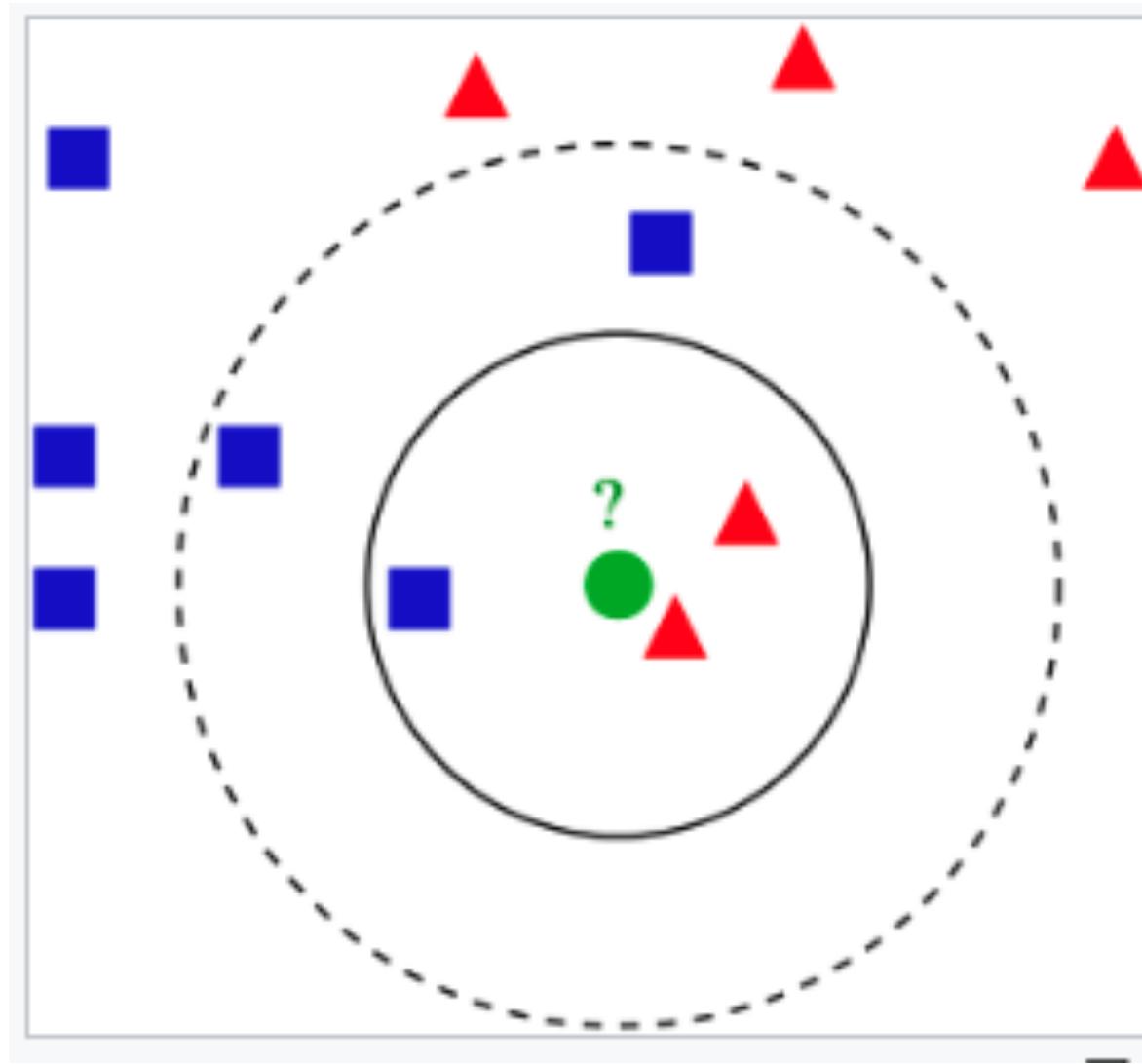


- ✓ Dramatic success of pre-trained LMs on many NLP tasks
- ✓ Model parameters pack a lot of knowledge
- ✓ Emergent abilities at massive scales.

- Lack of transparency into model mechanisms
- Hard to update/add new knowledge
- Less controllable

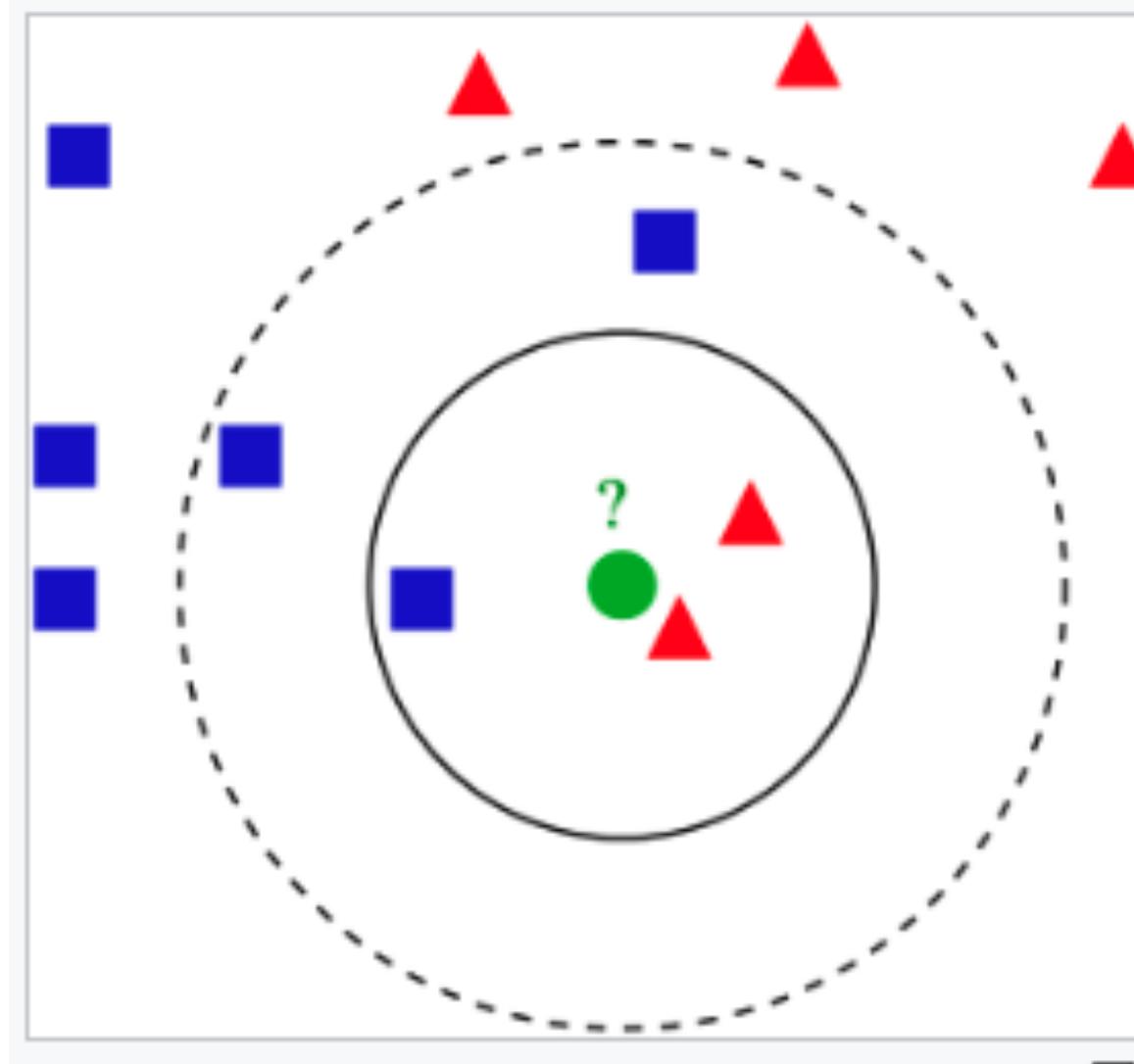
# Nonparametric NLP Models

# Nonparametric NLP Models



K-nearest neighbor classification  
(Image from Wikipedia)

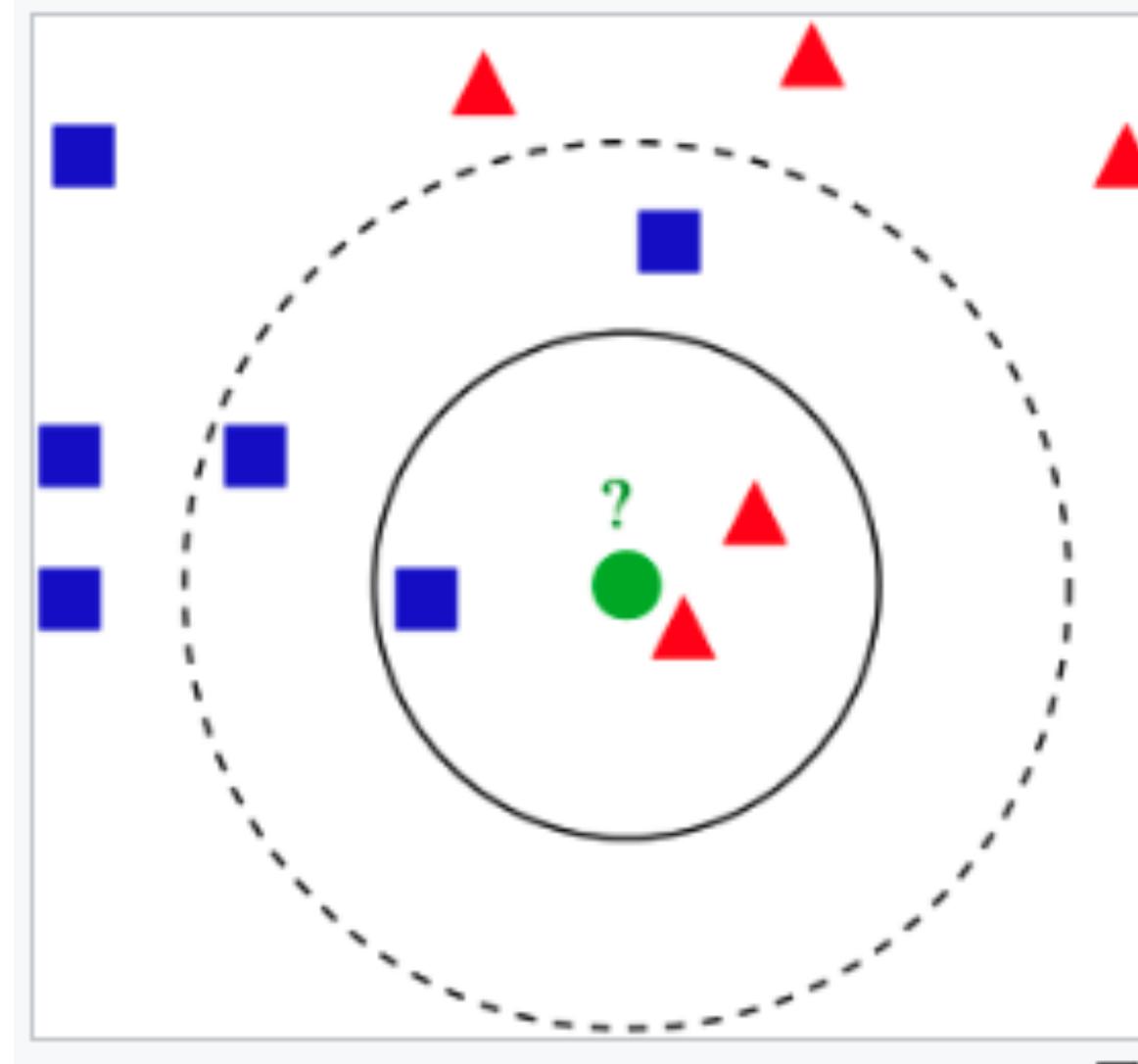
# Nonparametric NLP Models



K-nearest neighbor classification  
(Image from Wikipedia)

- ✓ Addition and deletion of data is easy!
- ✓ Model prediction explained by observing nearest neighbor
- ✓ Controllable - a prediction gone wrong can be fixed by adding more nearest neighbors

# Nonparametric NLP Models



K-nearest neighbor classification  
(Image from Wikipedia)

- ✓ Addition and deletion of data is easy!
- ✓ Model prediction explained by observing nearest neighbor
- ✓ Controllable - a prediction gone wrong can be fixed by adding more nearest neighbors

- Much less accurate than parametric models



# Semiparametric Models

# Semiparametric Models

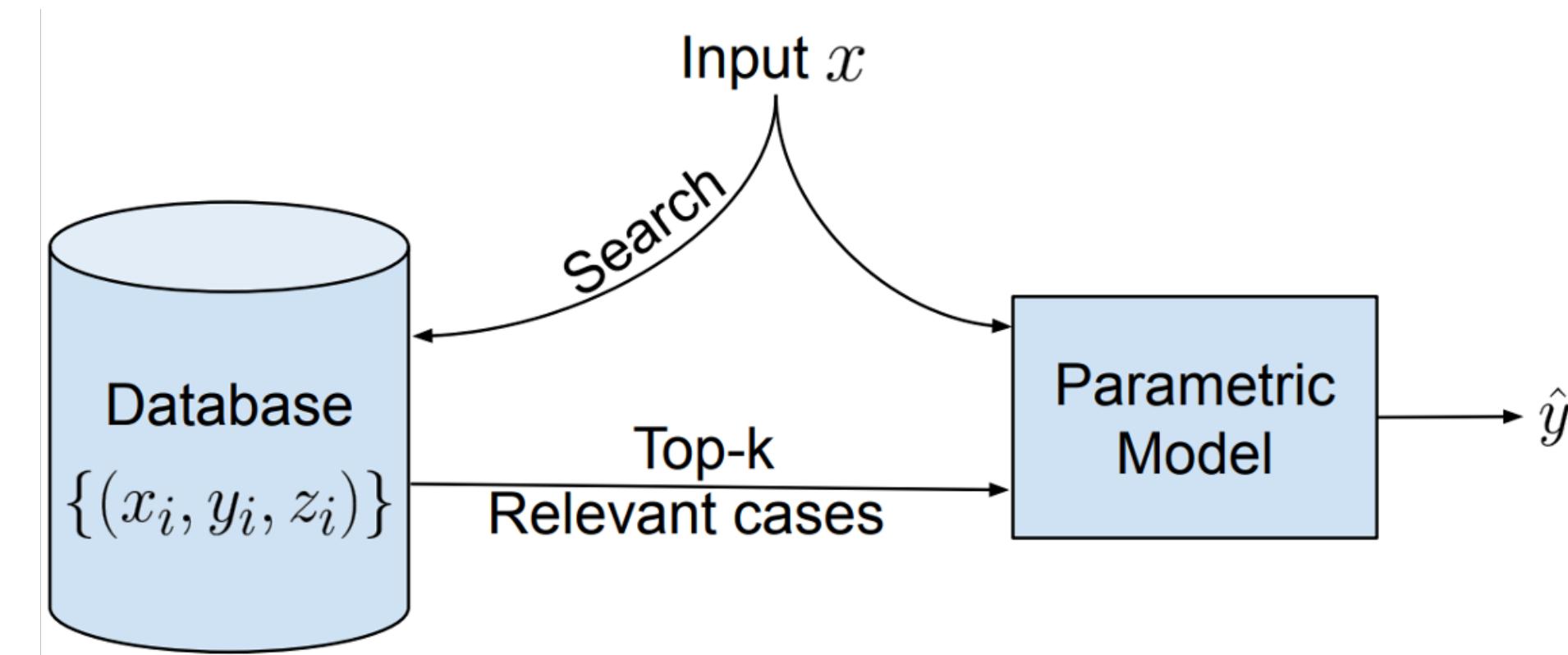
From Wikipedia, the free encyclopedia

In statistics, a **semiparametric model** is a **statistical model** that has **parametric** and **nonparametric** components.

# Semiparametric Models

From Wikipedia, the free encyclopedia

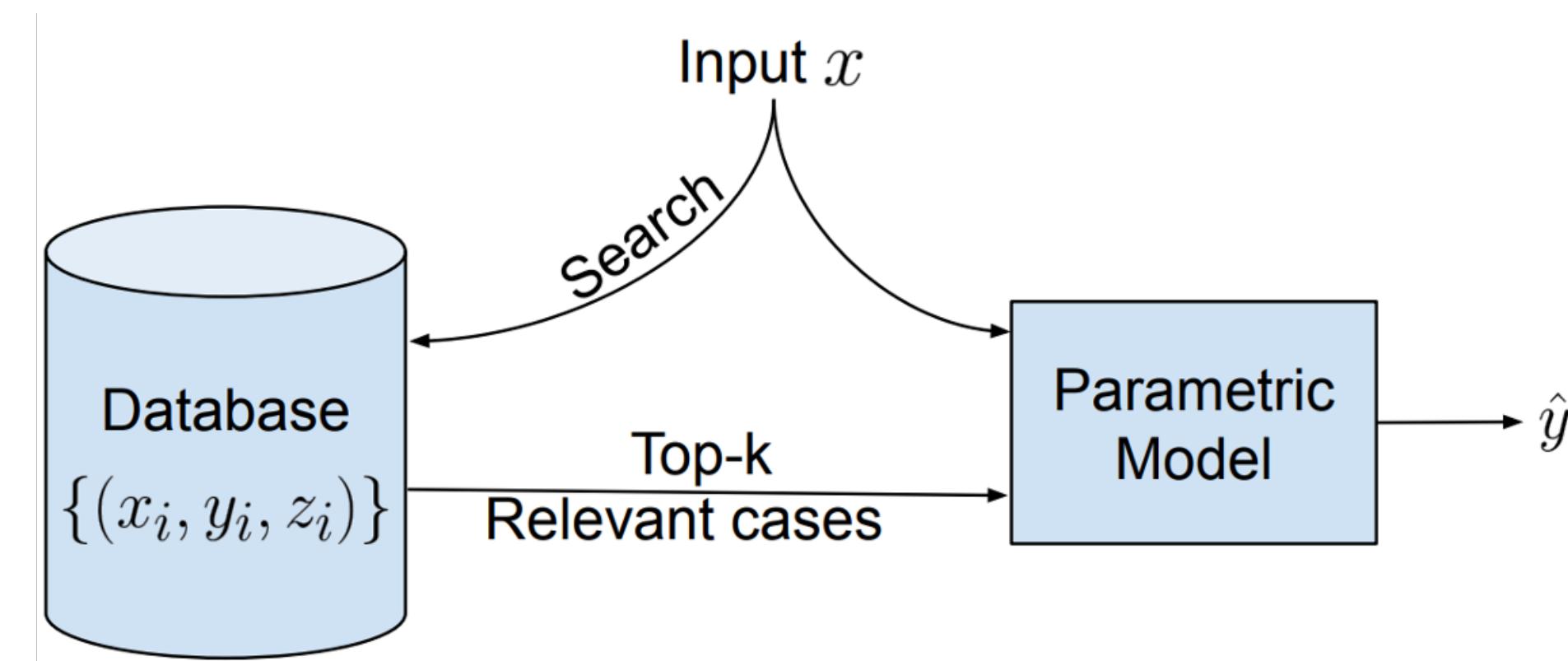
In statistics, a **semiparametric model** is a **statistical model** that has **parametric** and **nonparametric** components.



# Semiparametric Models

From Wikipedia, the free encyclopedia

In statistics, a **semiparametric model** is a **statistical model** that has **parametric** and **nonparametric** components.

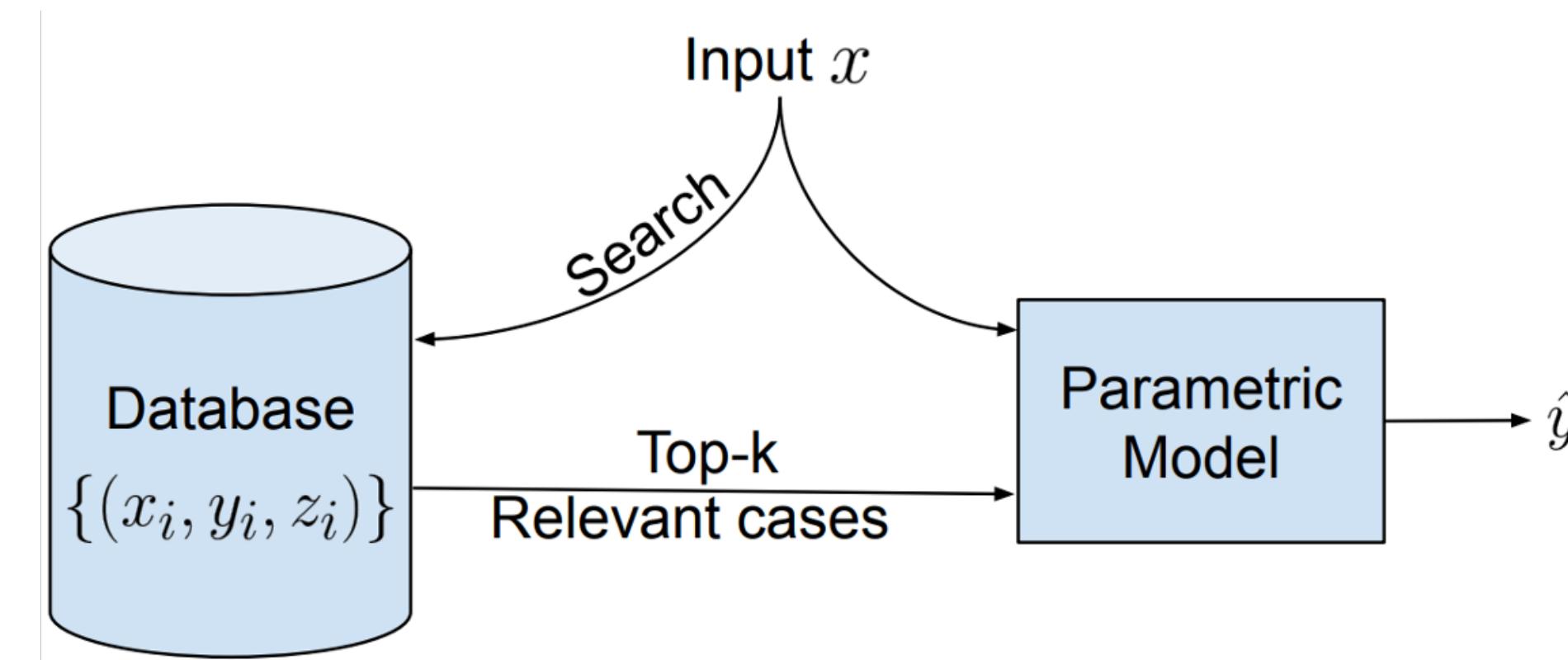


- ◆ Parametric component —> encodes logic
- ◆ Nonparametric component —> retrieves knowledge

# Semiparametric Models

From Wikipedia, the free encyclopedia

In statistics, a **semiparametric model** is a **statistical model** that has **parametric** and **nonparametric** components.



- ◆ Parametric component —> encodes logic
- ◆ Nonparametric component —> retrieves knowledge

Question Answering (Chen et al 2017, Karpukhin et al 2020, Lewis et al 2021, etc)  
Machine Translation (Gu et al 2018, Khandelwal et al 2021)  
Language Modeling (Lee et al 2019, Khandelwal et al 2020)  
Semantic Parsing (Das et al 2021)  
Protein Structure Prediction (Jumper et al 2021)

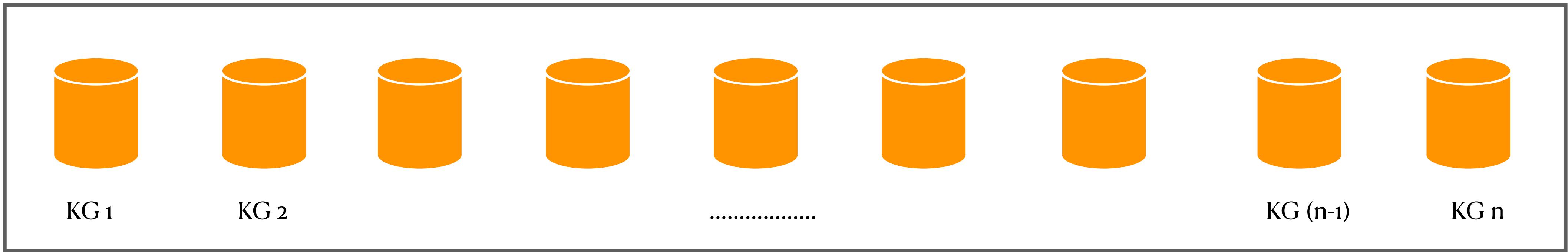
# Challenges in Building NL Interfaces

# Challenges in Building NL Interfaces

- ♦ In an enterprise, plethora of KGs (each with their own schema), created by each team.

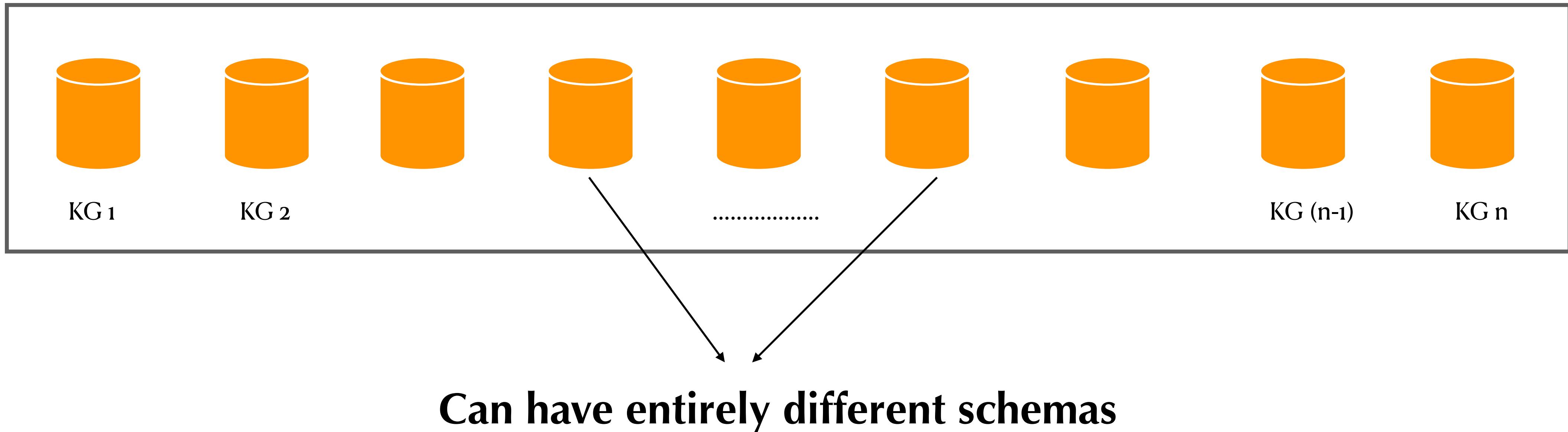
# Challenges in Building NL Interfaces

- ♦ In an enterprise, plethora of KGs (each with their own schema), created by each team.



# Challenges in Building NL Interfaces

- ♦ In an enterprise, plethora of KGs (each with their own schema), created by each team.



# Challenges in Building NL Interfaces

- ♦ In an enterprise, plethora of KGs (each with their own schema), created by each team.

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions

What are the top 5 countries by GDP that have a free trade agreement with the European Union, and what are the key industries that drive their economic growth?

Which countries have the highest percentage of their electricity generation coming from renewable sources?

# Challenges in Building NL Interfaces

- ♦ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ♦ Complex Questions

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions
- ◆ Lack of training data

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions
- ◆ Lack of training data
  - ◆ Need experts to annotate, hence expensive

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions
- ◆ Lack of training data
  - ◆ Need experts to annotate, hence expensive
  - ◆ For broad KGs (e.g. Wikidata) hard to obtain coverage for all query types

# Challenges in Building NL Interfaces

- ◆ In an enterprise, plethora of KGs (each with their own schema), created by each team.
- ◆ Complex Questions
- ◆ Lack of training data
  - ◆ Need experts to annotate, hence expensive
  - ◆ For broad KGs (e.g. Wikidata) hard to obtain coverage for all query types
  - ◆ Privacy concerns sometimes does not allow data collection.

# **Desiderata for a NL interface for KGs**

# Desiderata for a NL interface for KGs

- ◆ Build a natural language interface for querying KGs that

# Desiderata for a NL interface for KGs

- ◆ Build a natural language interface for querying KGs that
  - ◆ Works for *Any Knowledge Graph* (irrespective of their schema)

# Desiderata for a NL interface for KGs

- ◆ Build a natural language interface for querying KGs that
  - ◆ Works for *Any Knowledge Graph* (irrespective of their schema)
  - ◆ Requires *no training data* (i.e. completely unsupervised)

# Desiderata for a NL interface for KGs

- ◆ Build a natural language interface for querying KGs that
  - ◆ Works for *Any Knowledge Graph* (irrespective of their schema)
  - ◆ Requires *no training data* (i.e. completely unsupervised)
  - ◆ Ready to be used in *reasonable time* (~1 day)

# **Key Difference from Prior Work**

# Key Difference from Prior Work

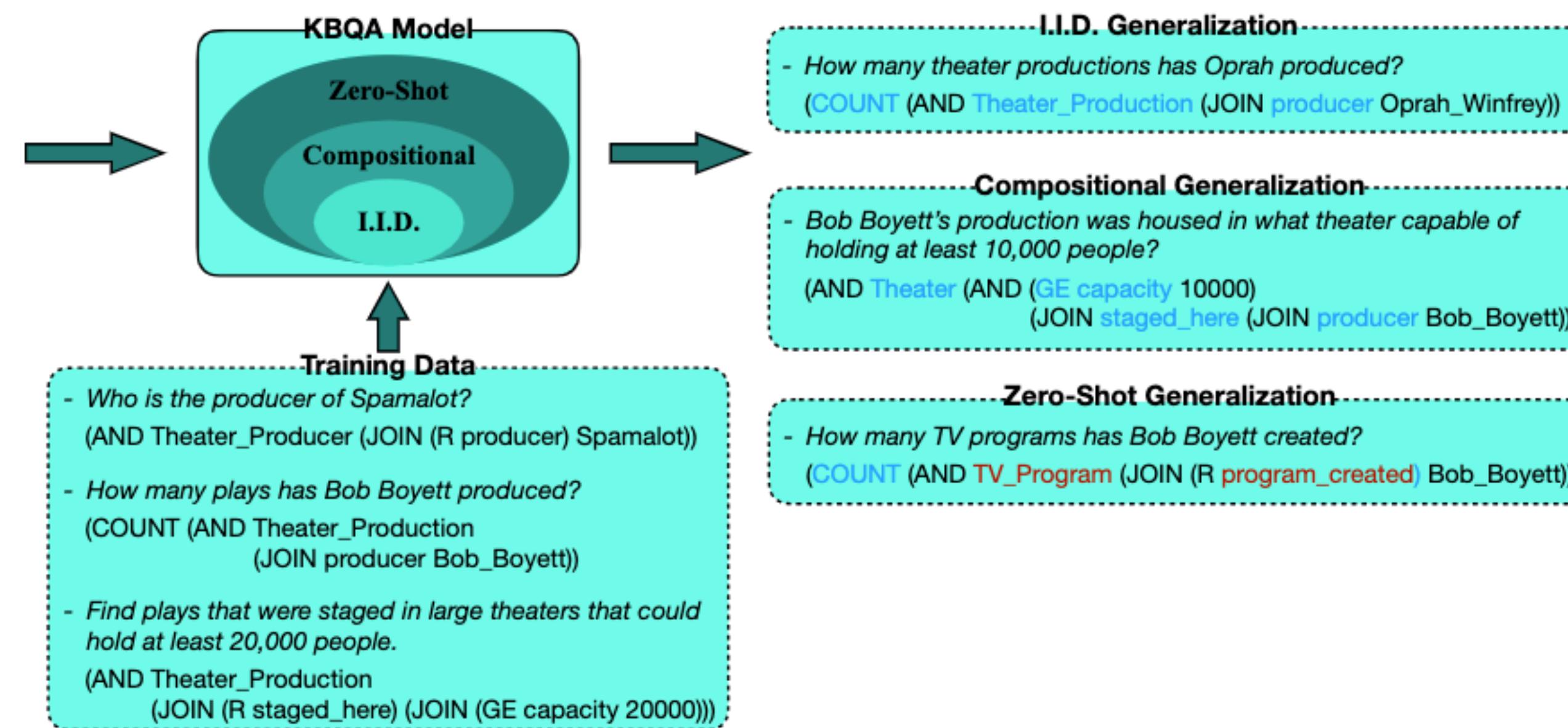
- ♦ Zero-shot QA over KGs is not new (Su et al 2021)

# Key Difference from Prior Work

- ♦ Zero-shot QA over KGs is not new (Su et al 2021)
- ♦ However prior work assumes availability of *some training data* and queries that need a single unseen schema item were considered as zero-shot.

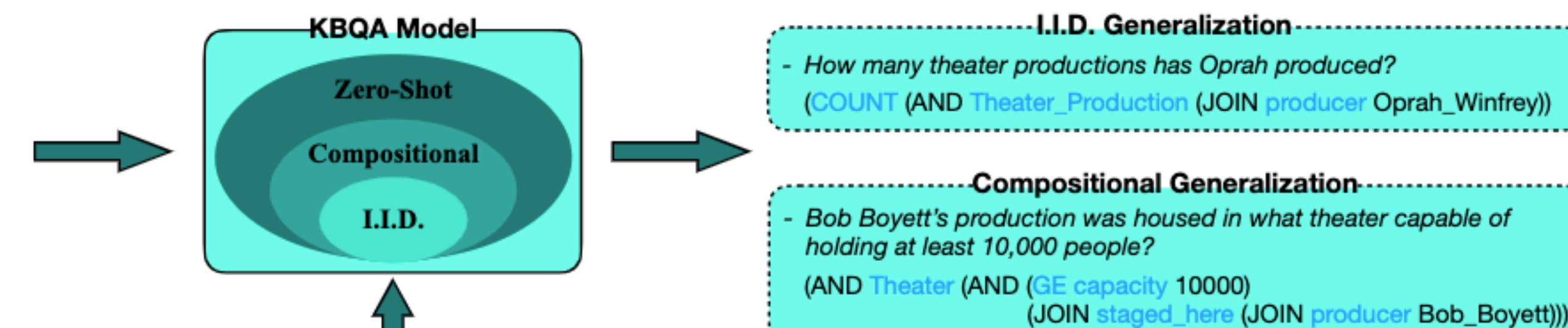
# Key Difference from Prior Work

- ◆ Zero-shot QA over KGs is not new (Su et al 2021)
- ◆ However prior work assumes availability of *some training data* and queries that need a single unseen schema item were considered as zero-shot.



# Key Difference from Prior Work

- ◆ Zero-shot QA over KGs is not new (Su et al 2021)
- ◆ However prior work assumes availability of *some training data* and queries that need a single unseen schema item were considered as zero-shot.



No training questions; no prior knowledge of KG schema and query distribution.

- Find plays that were staged in large theaters that could hold at least 20,000 people.  
(AND Theater\_Production  
(JOIN (R staged\_here) (JOIN (GE capacity 20000))))

# Our Approach: Bring Your Own KG

# BRING YOUR OWN KG: Self-Supervised Program Synthesis for Zero-Shot KGQA

**Dhruv Agarwal<sup>1,\*</sup>, Rajarshi Das<sup>2,†</sup>, Sopan Khosla<sup>2,†</sup>, Rashmi Gangadharaiah<sup>2</sup>**

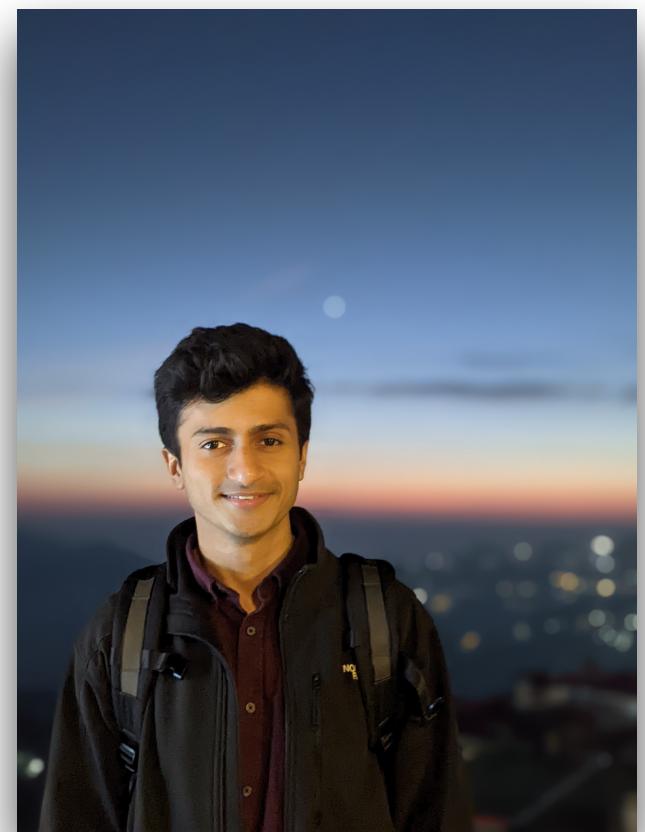
<sup>1</sup>University of Massachusetts Amherst, <sup>2</sup>AWS AI Labs

dagarwal@cs.umass.edu, {dasrajar, sopankh, rgangad}@amazon.com

## Abstract

We present BYOKG, a universal question-answering (QA) system that can operate on *any* knowledge graph (KG), requires no human-annotated training data, and can be ready to use within a day — attributes that are out-of-scope

of *some* training data (query-program pairs) ([Talmor and Berant, 2018](#); [Keysers et al., 2020](#); [Gu et al., 2021](#); [Dutt et al., 2023a](#); [Sen et al., 2023](#)), which, in practice, might be unrealistic. For example, in domains such as biomedicine and clinical decision-making, training data may be completely



# BRING YOUR OWN KG: Self-Supervised Program Synthesis for Zero-Shot KGQA

**Dhruv Agarwal<sup>1,\*</sup>, Rajarshi Das<sup>2,†</sup>, Sopan Khosla<sup>2,†</sup>, Rashmi Gangadharaiah<sup>2</sup>**

<sup>1</sup>University of Massachusetts Amherst, <sup>2</sup>AWS AI Labs  
dagarwal@cs.umass.edu, {dasrajar, sopankh, rgangad}@amazon.com

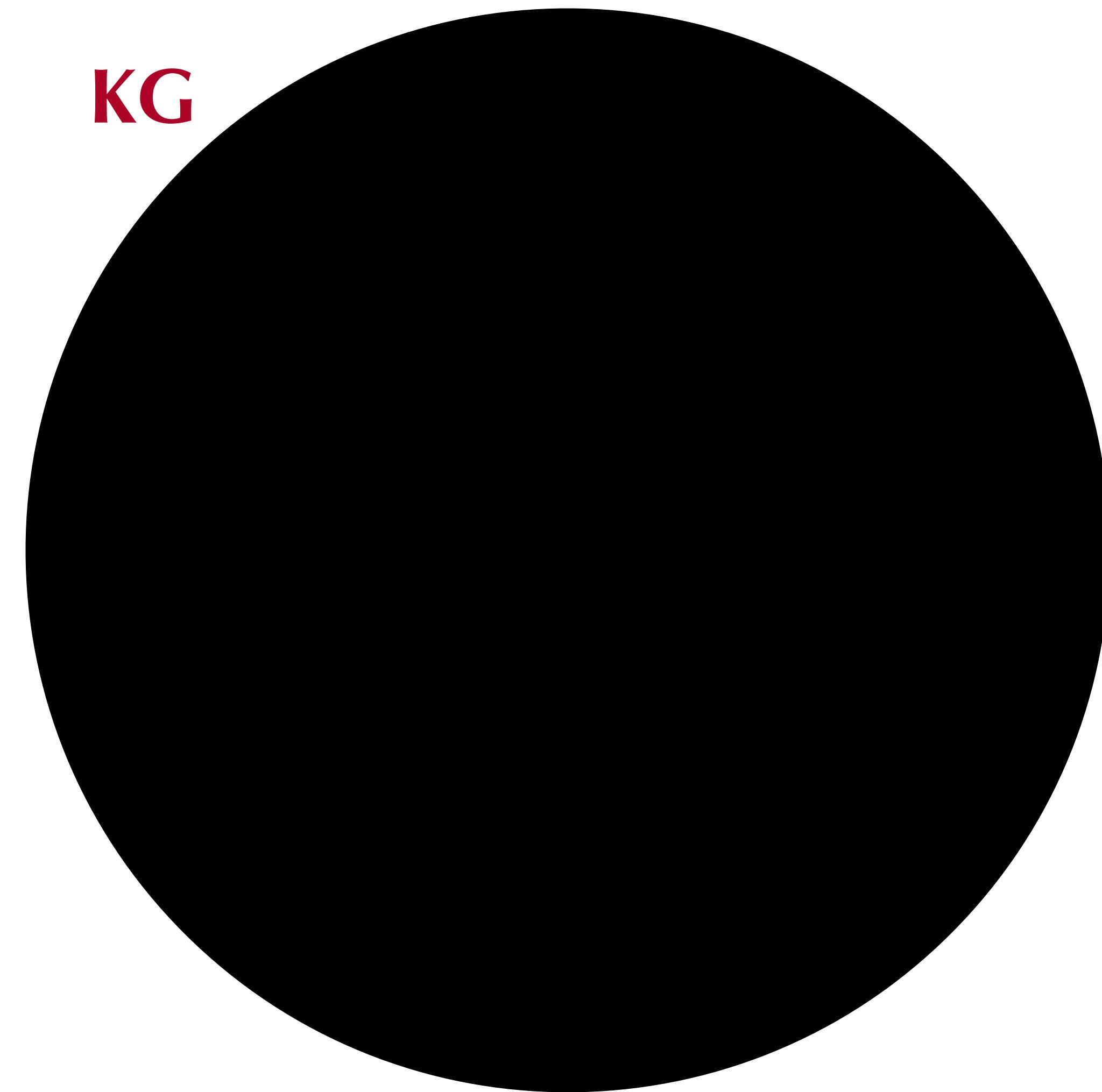
## Abstract

We present BYOKG, a universal question-answering (QA) system that can operate on *any* knowledge graph (KG), requires no human-annotated training data, and can be ready to use within a day — attributes that are out-of-scope

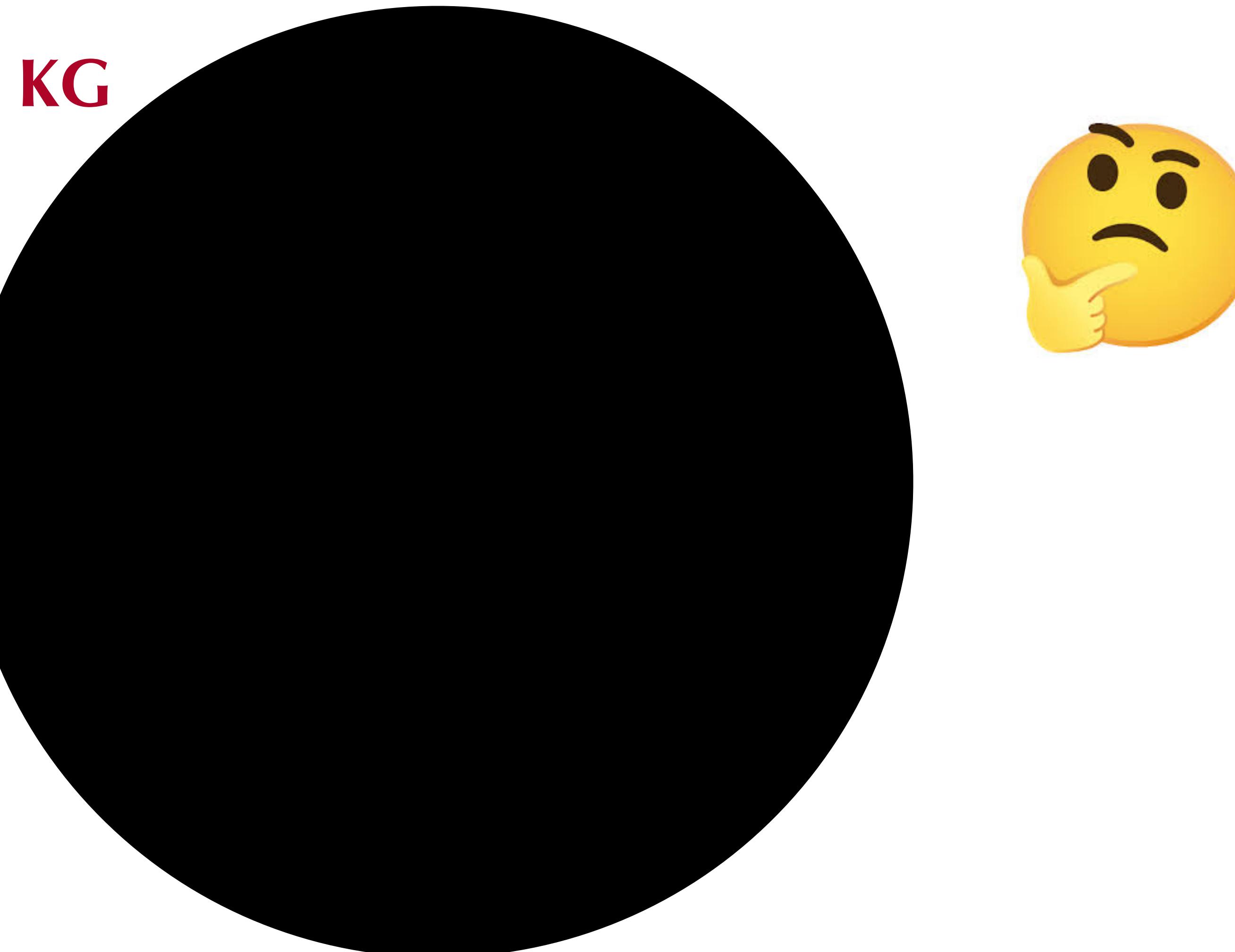
of *some* training data (query-program pairs) ([Talmor and Berant, 2018](#); [Keysers et al., 2020](#); [Gu et al., 2021](#); [Dutt et al., 2023a](#); [Sen et al., 2023](#)), which, in practice, might be unrealistic. For example, in domains such as biomedicine and clinical decision-making, training data may be completely

# How might a *Human* approach this?

# How might a *Human* approach this?

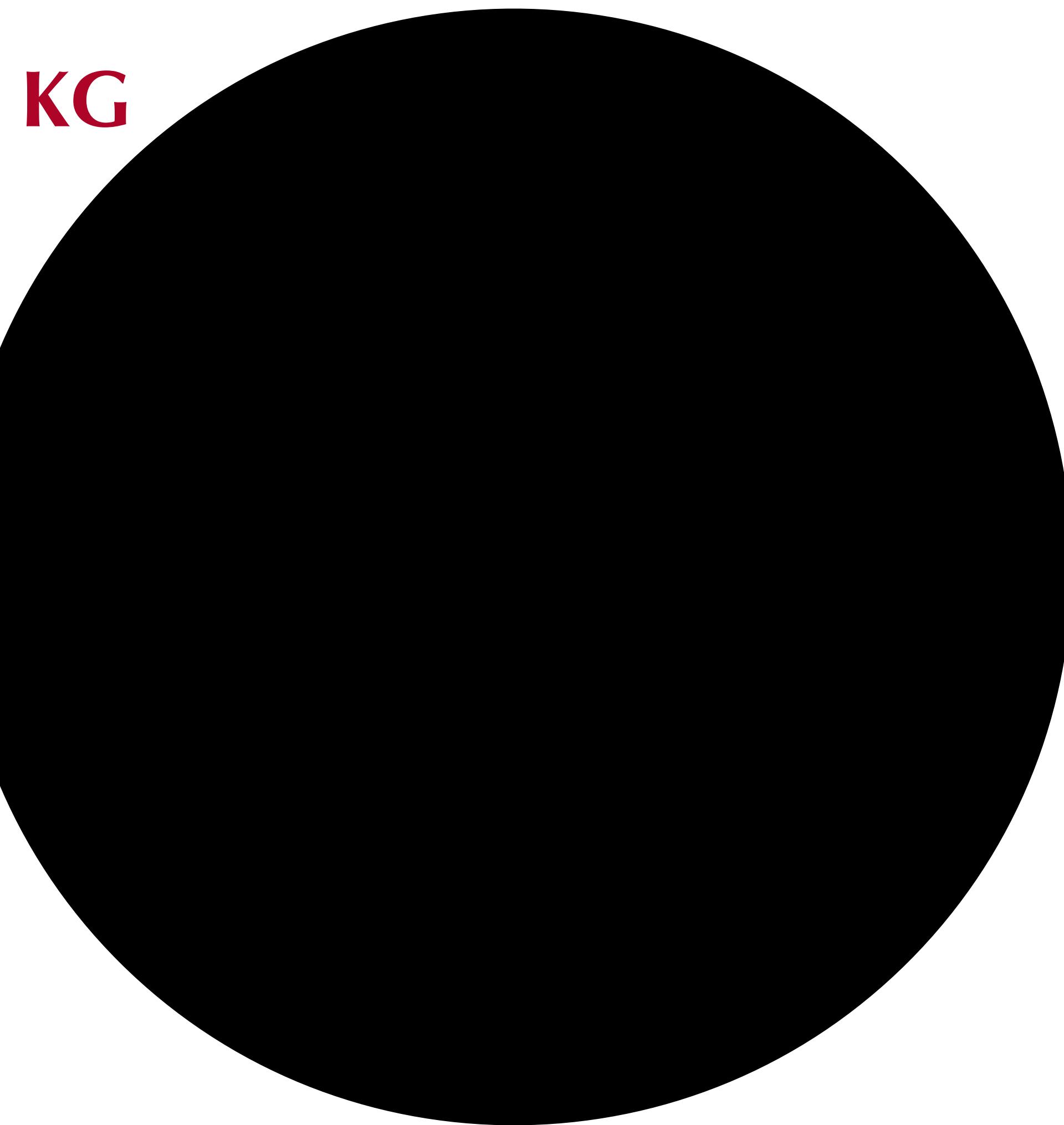


# How might a *Human* approach this?

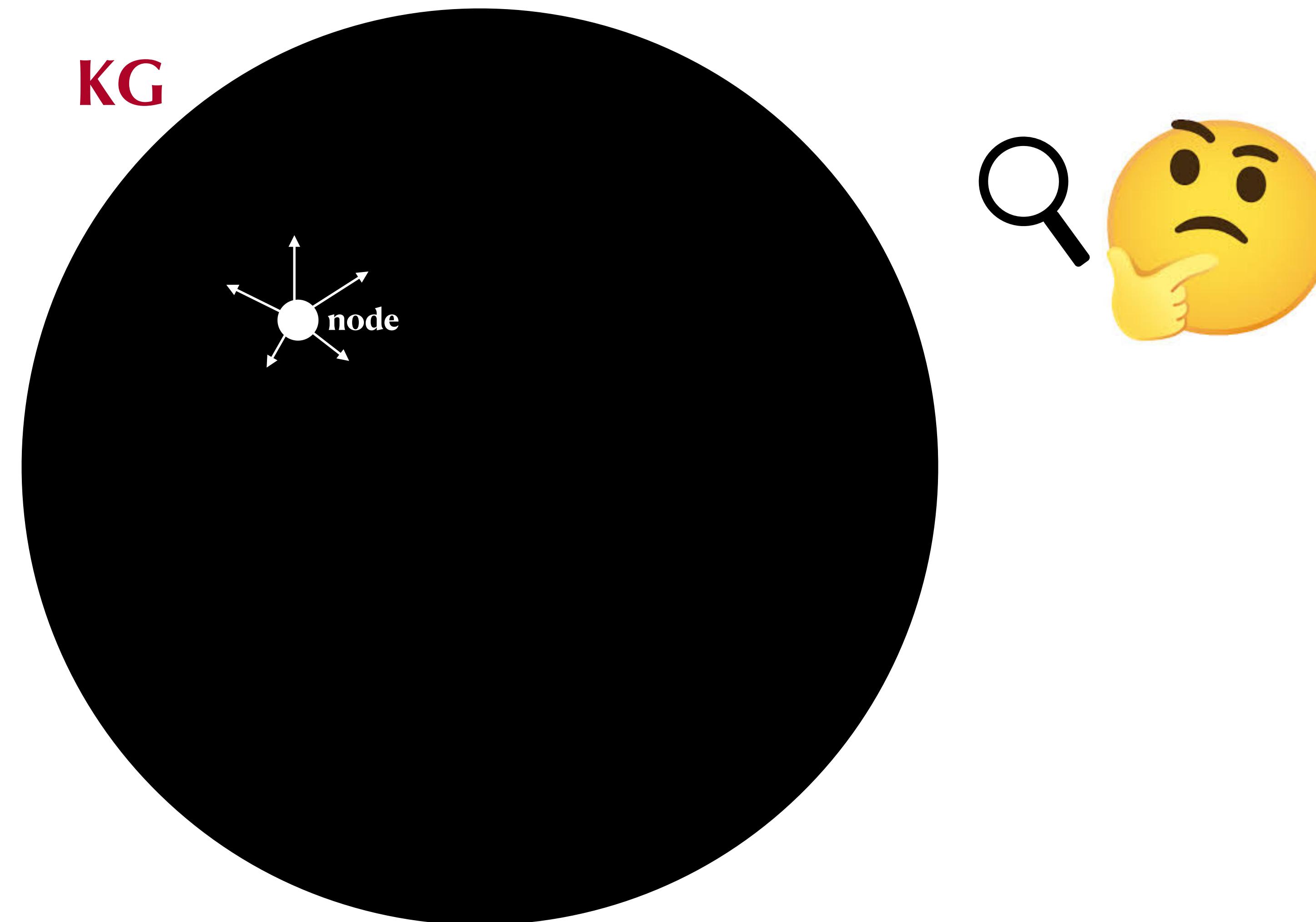


**KG**

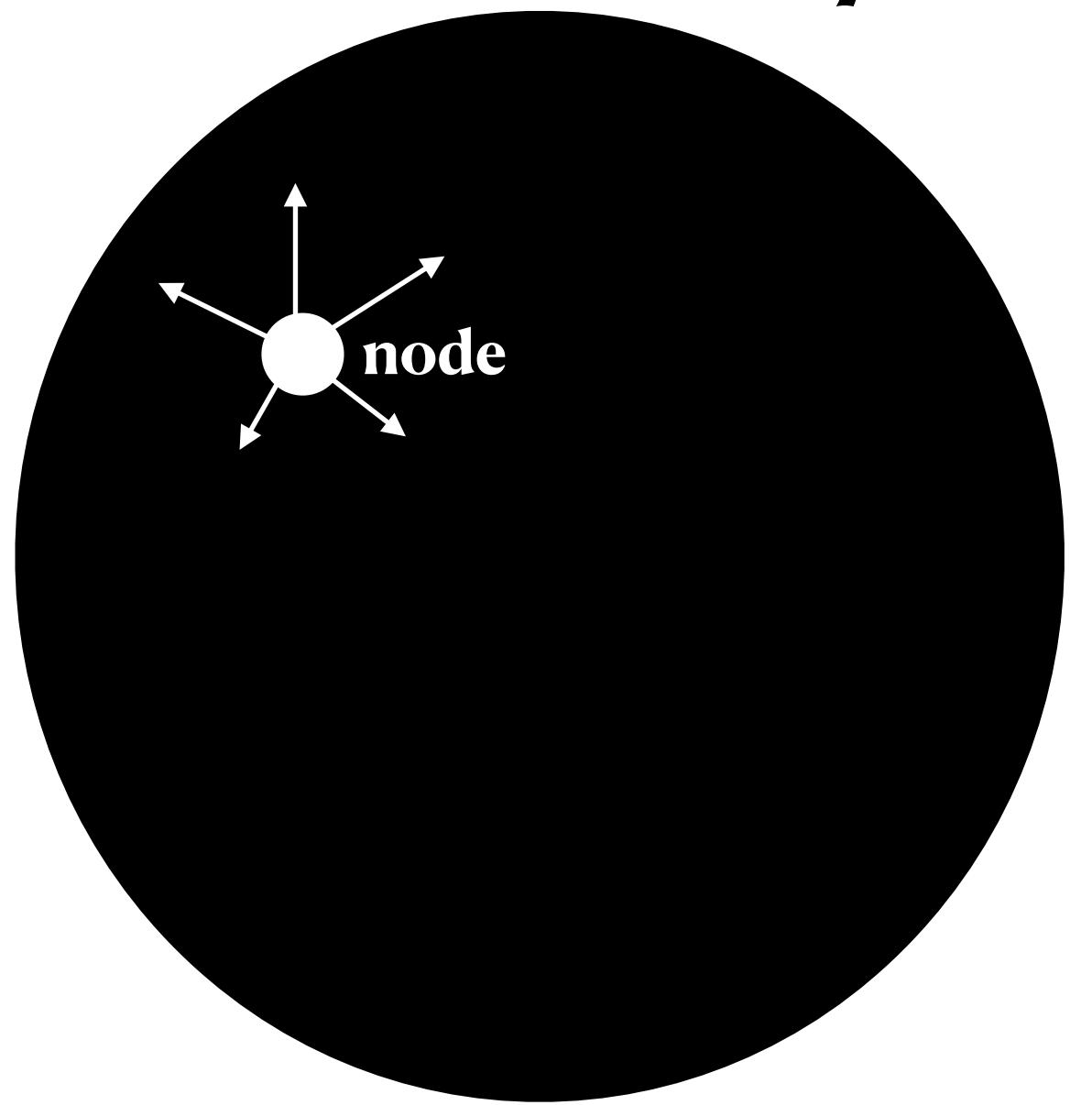
# Curiosity!



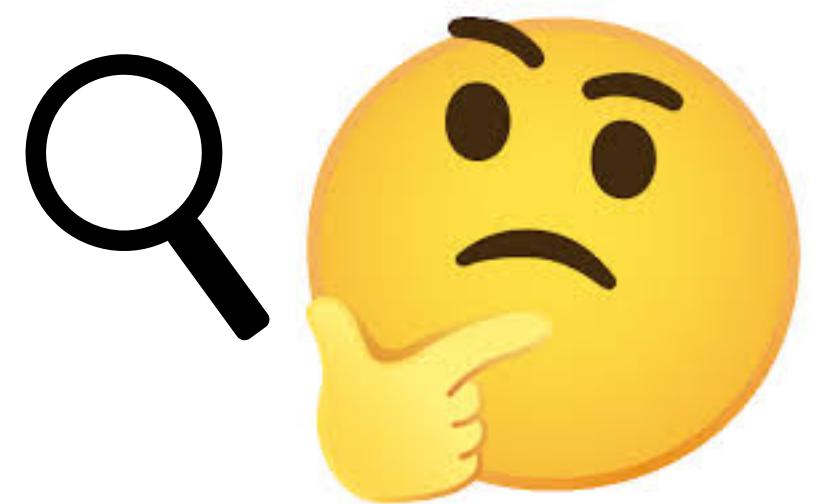
# Curiosity!



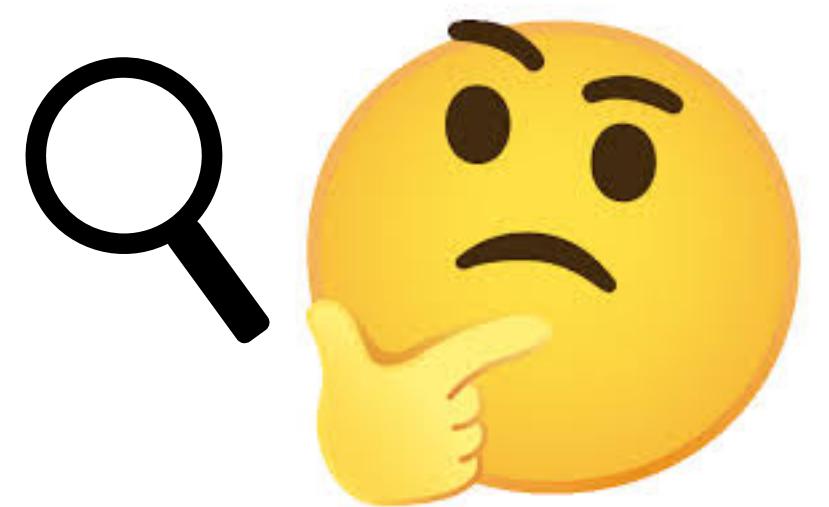
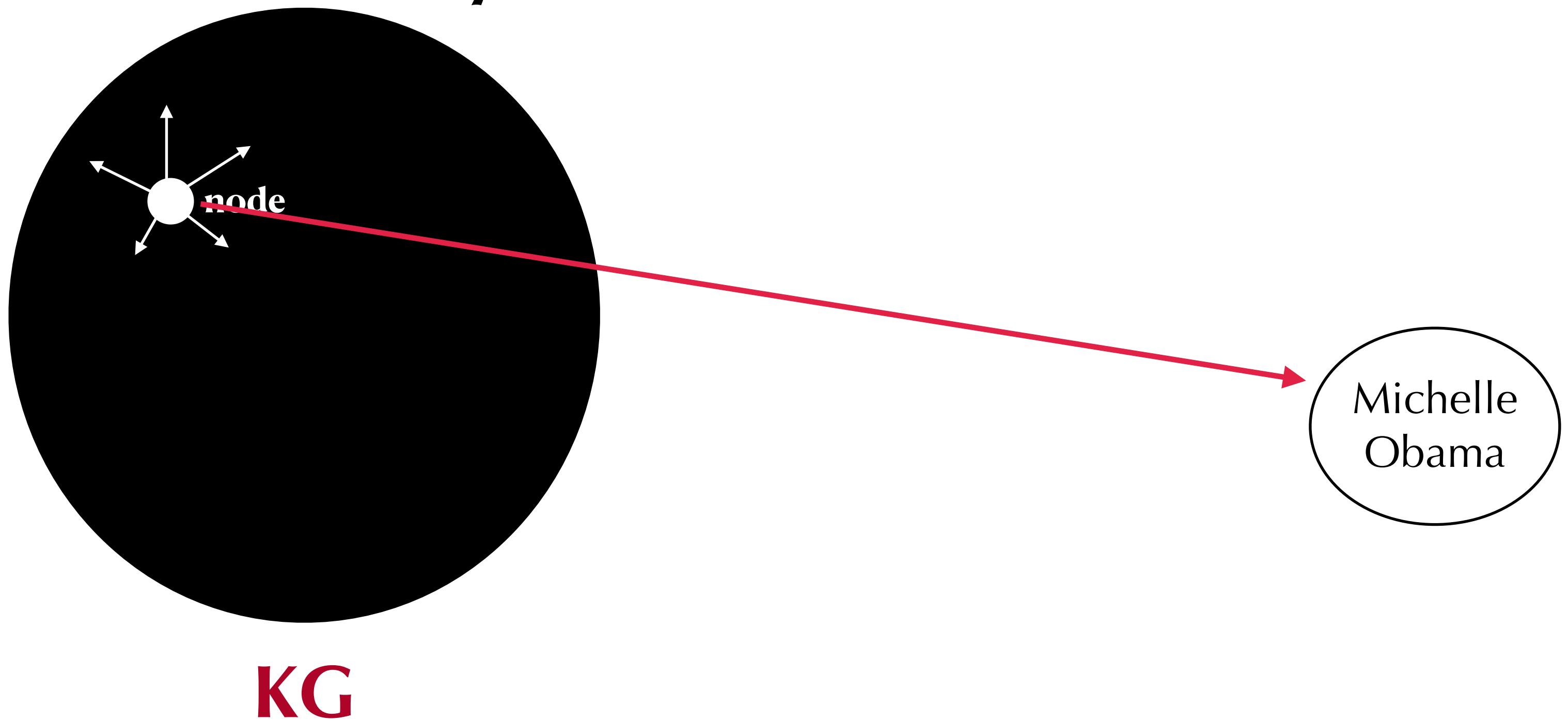
# Curiosity!



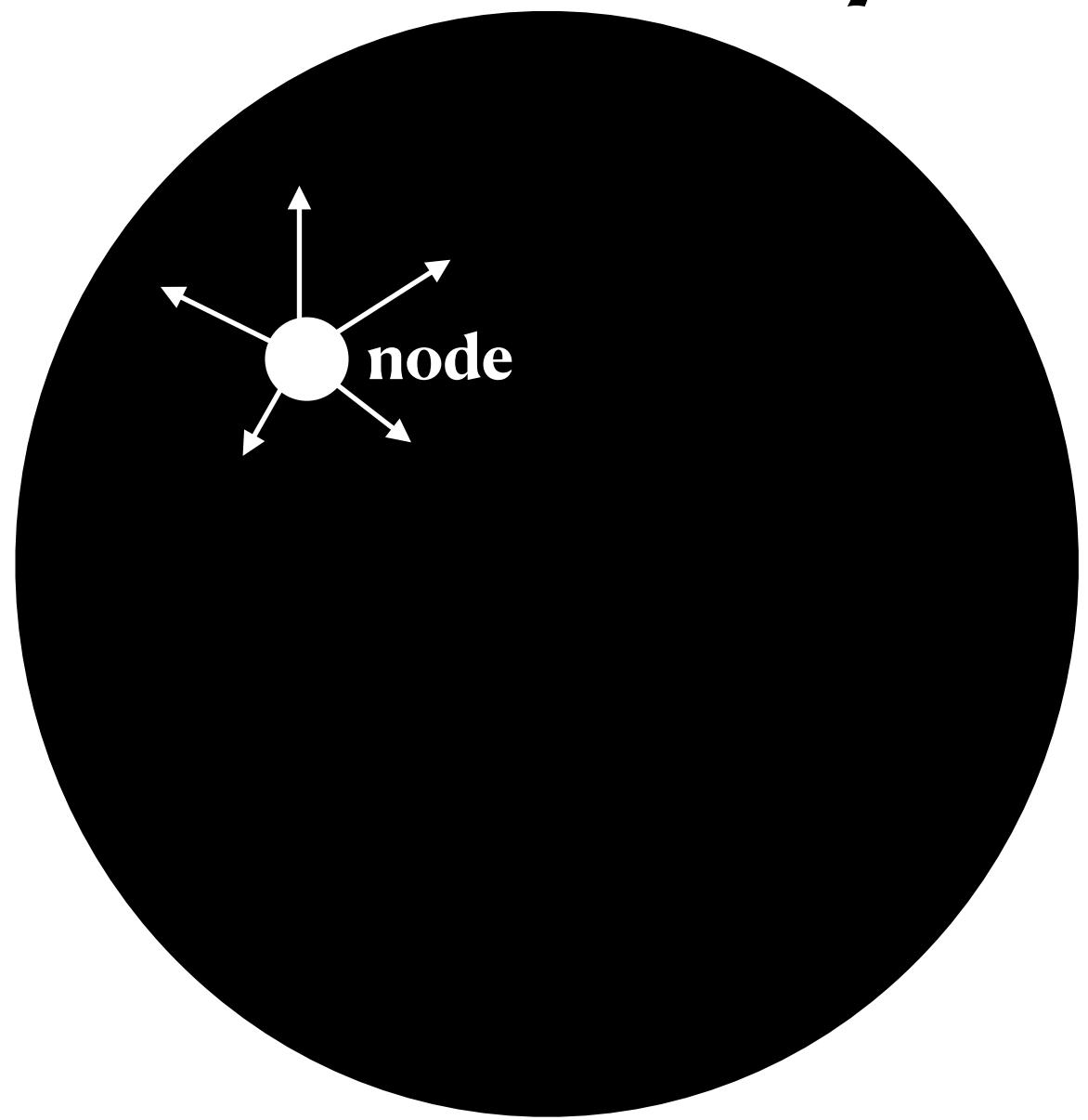
**KG**



# Curiosity!

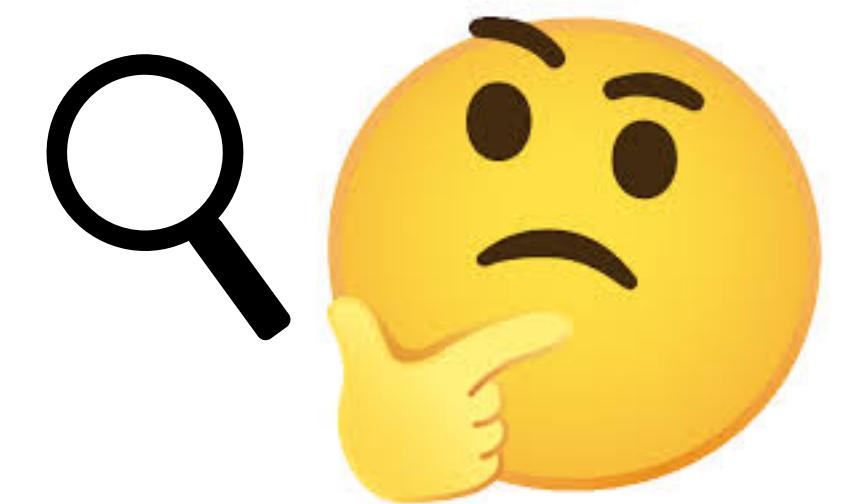


# Curiosity!

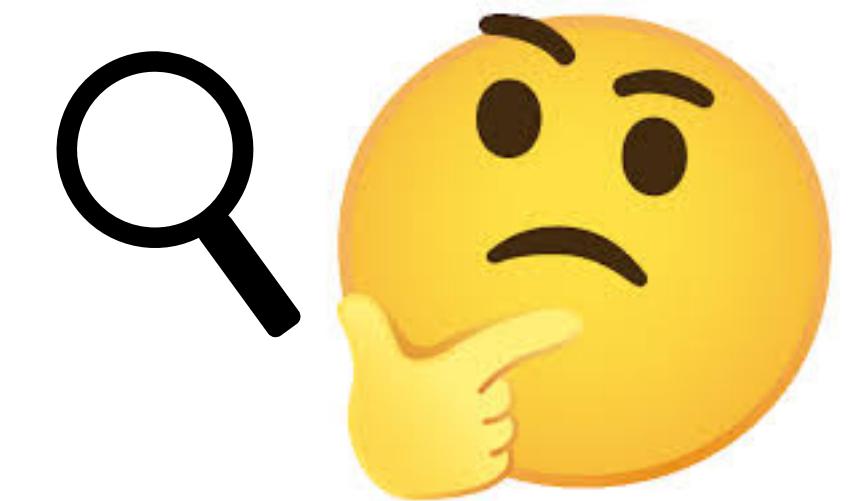
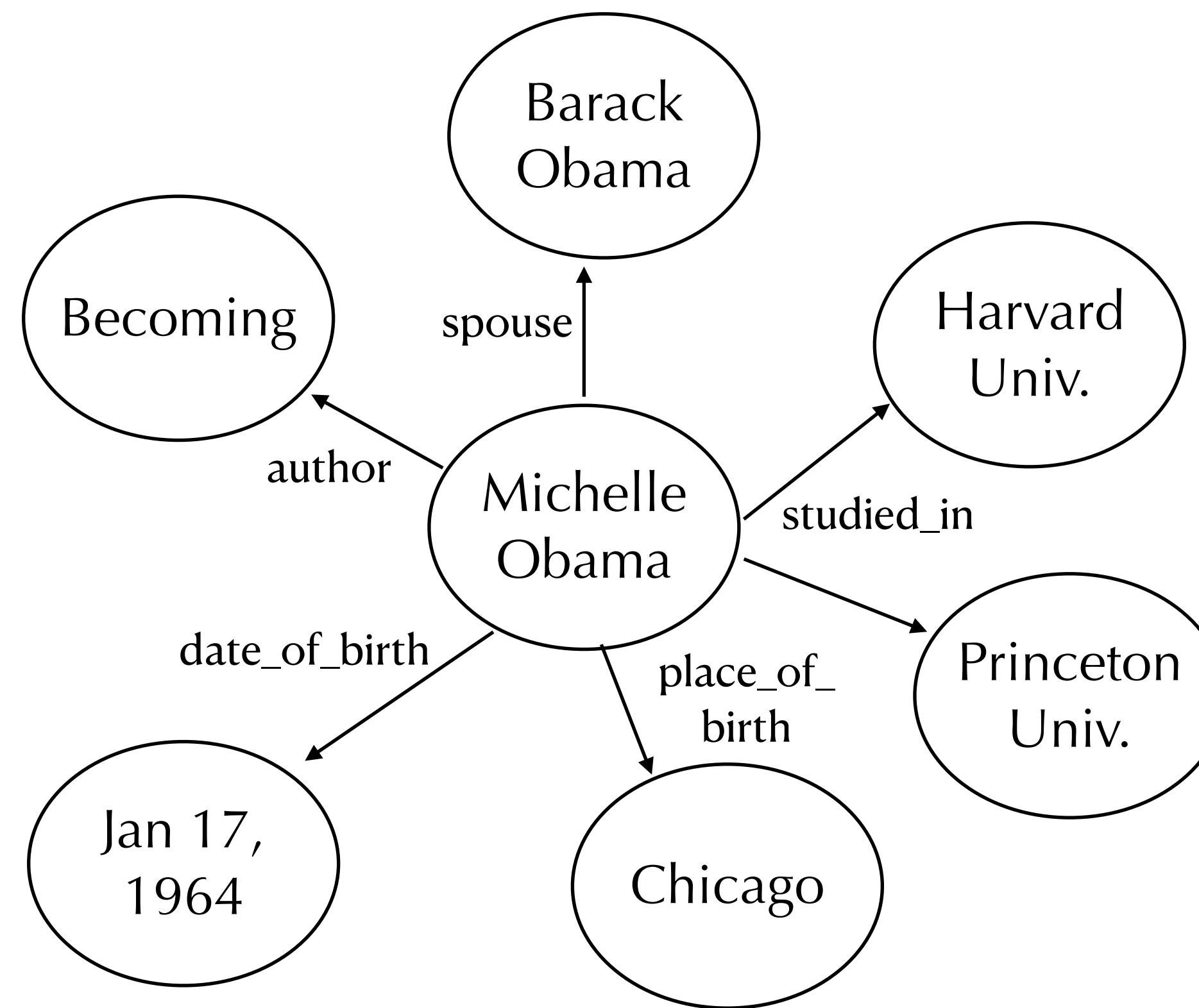
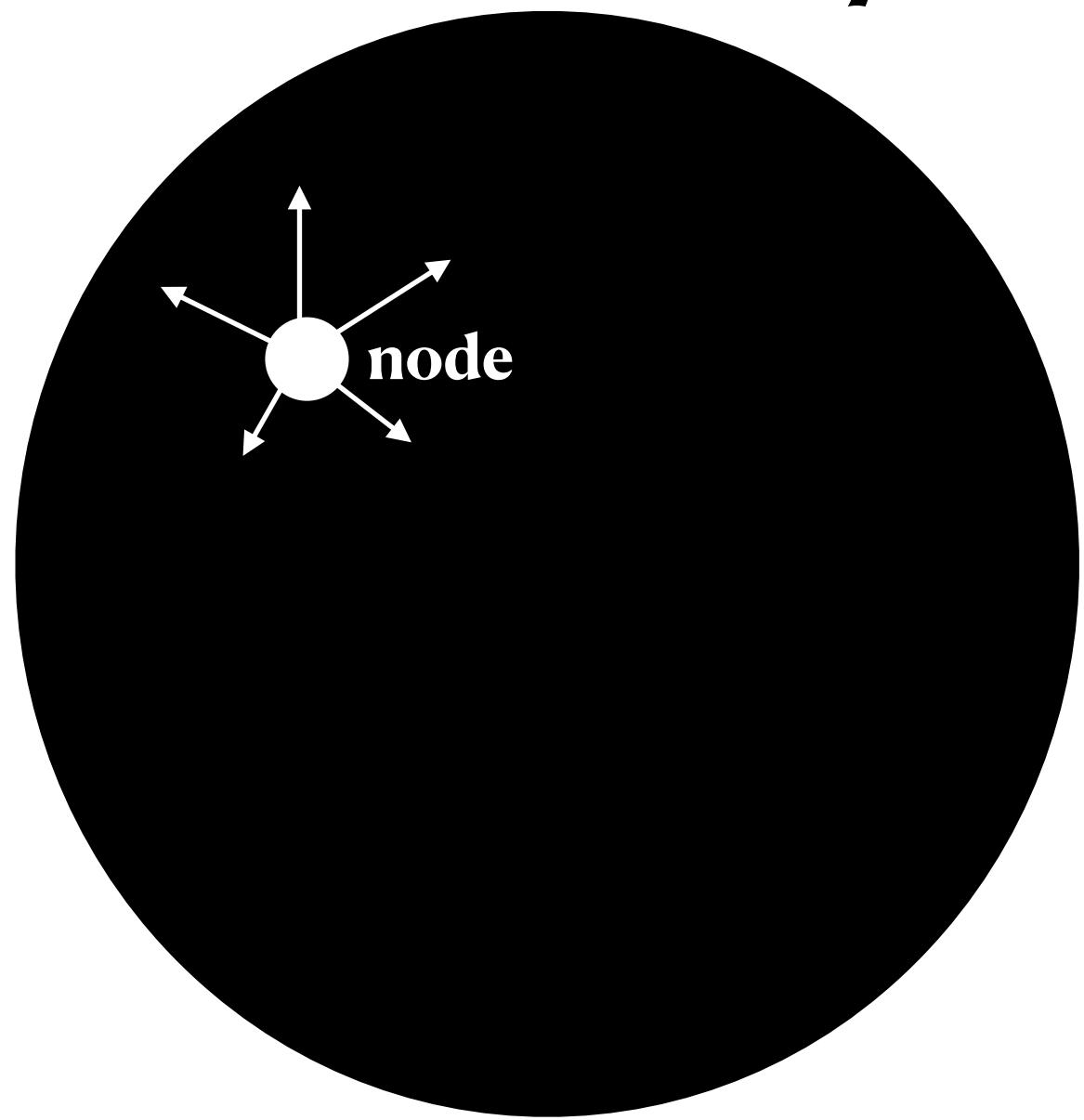


**KG**

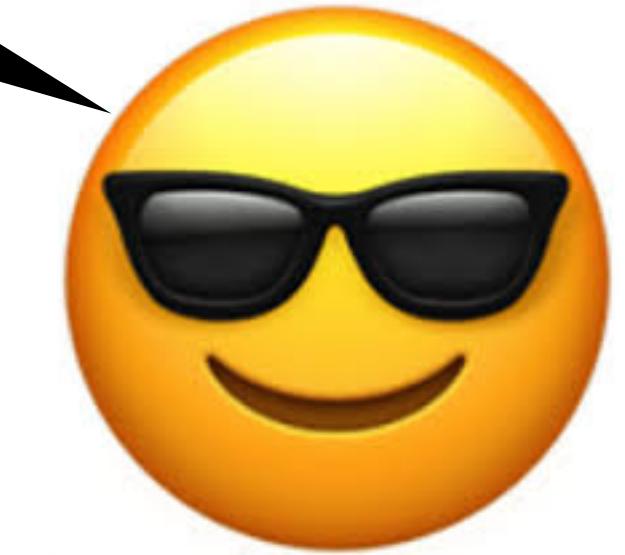
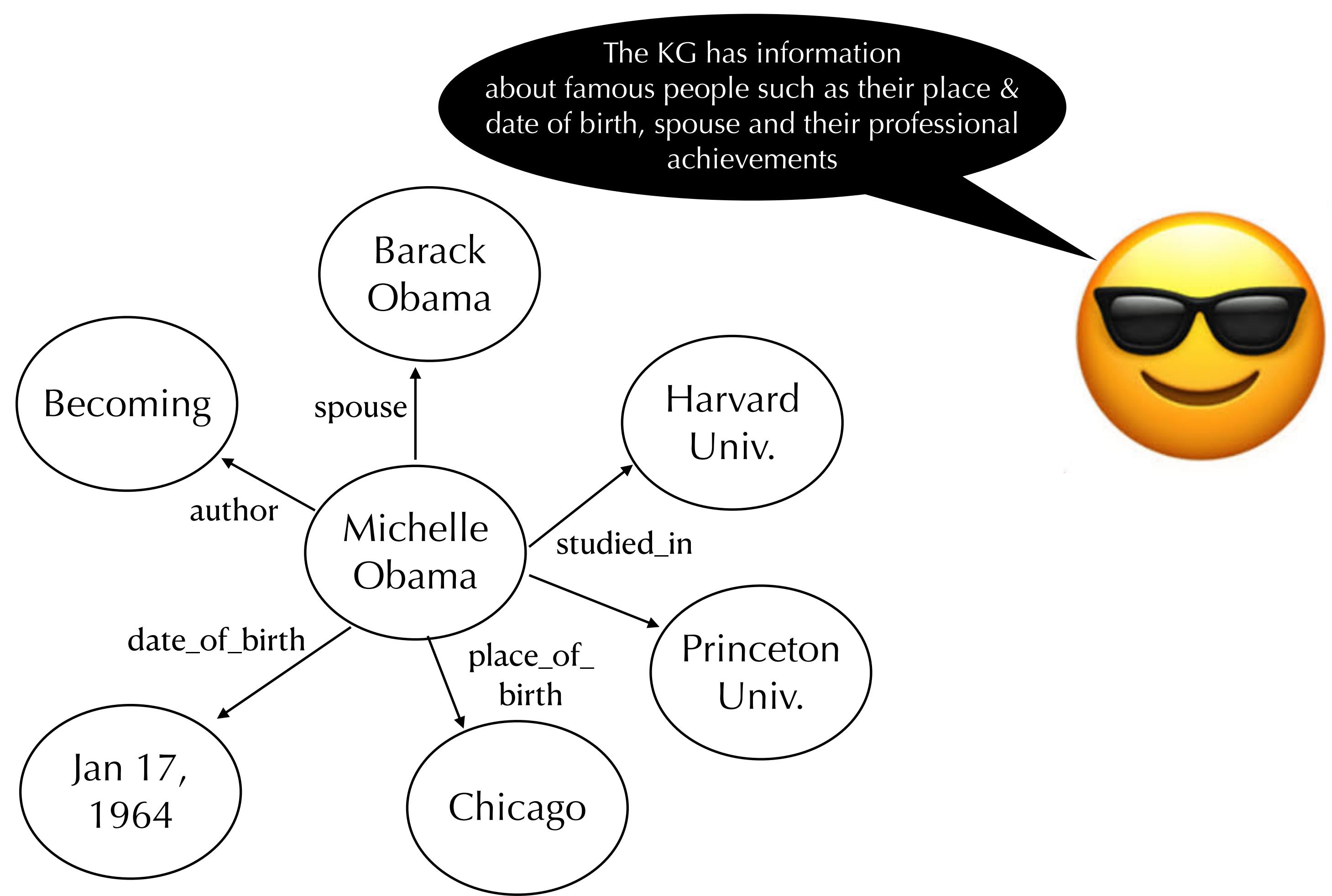
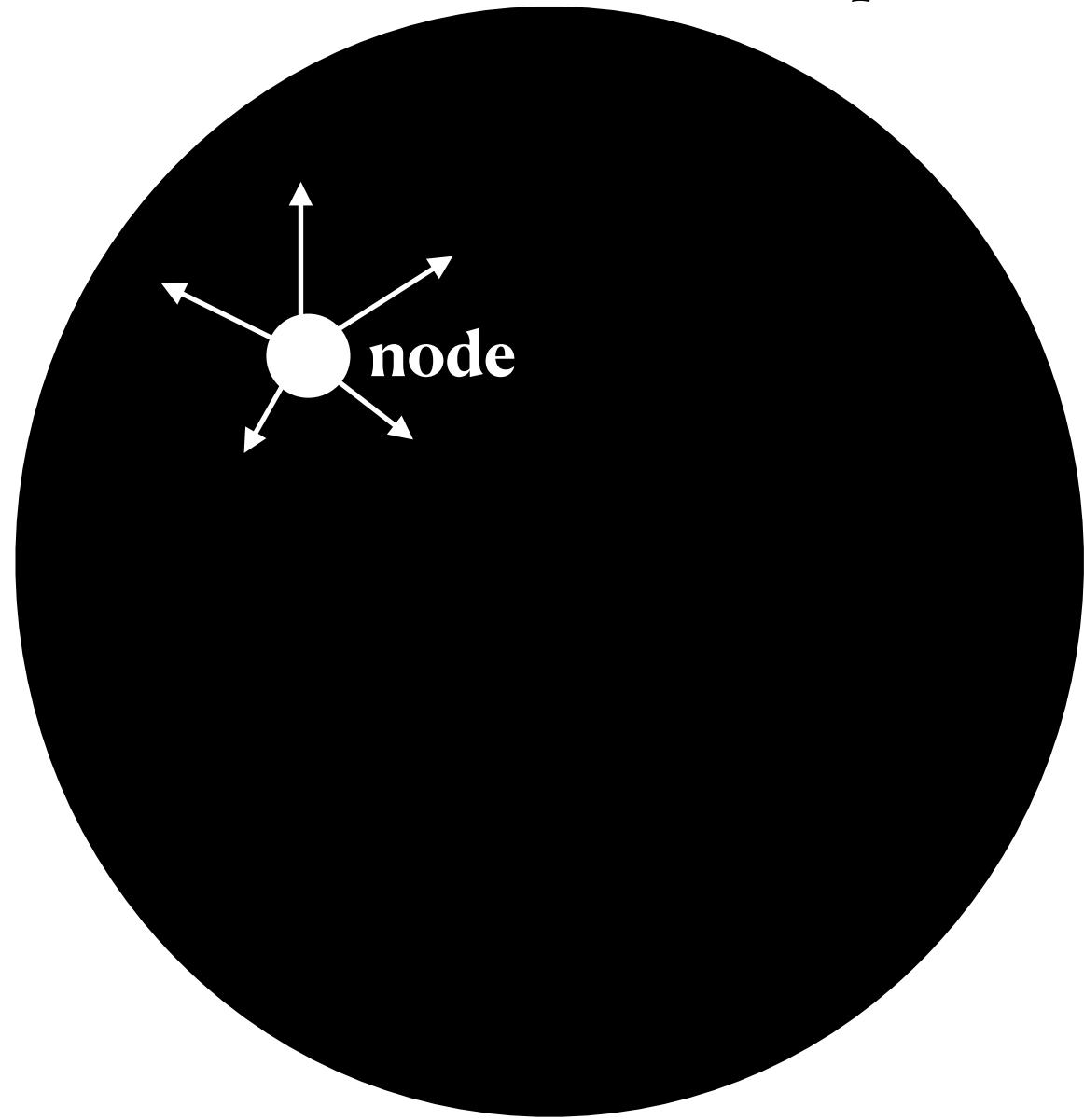
Michelle  
Obama



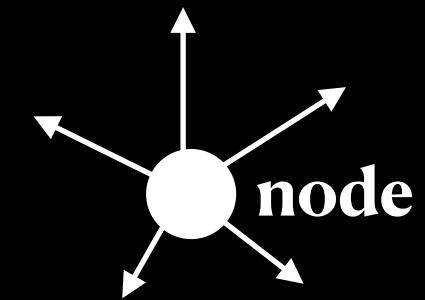
# Curiosity!



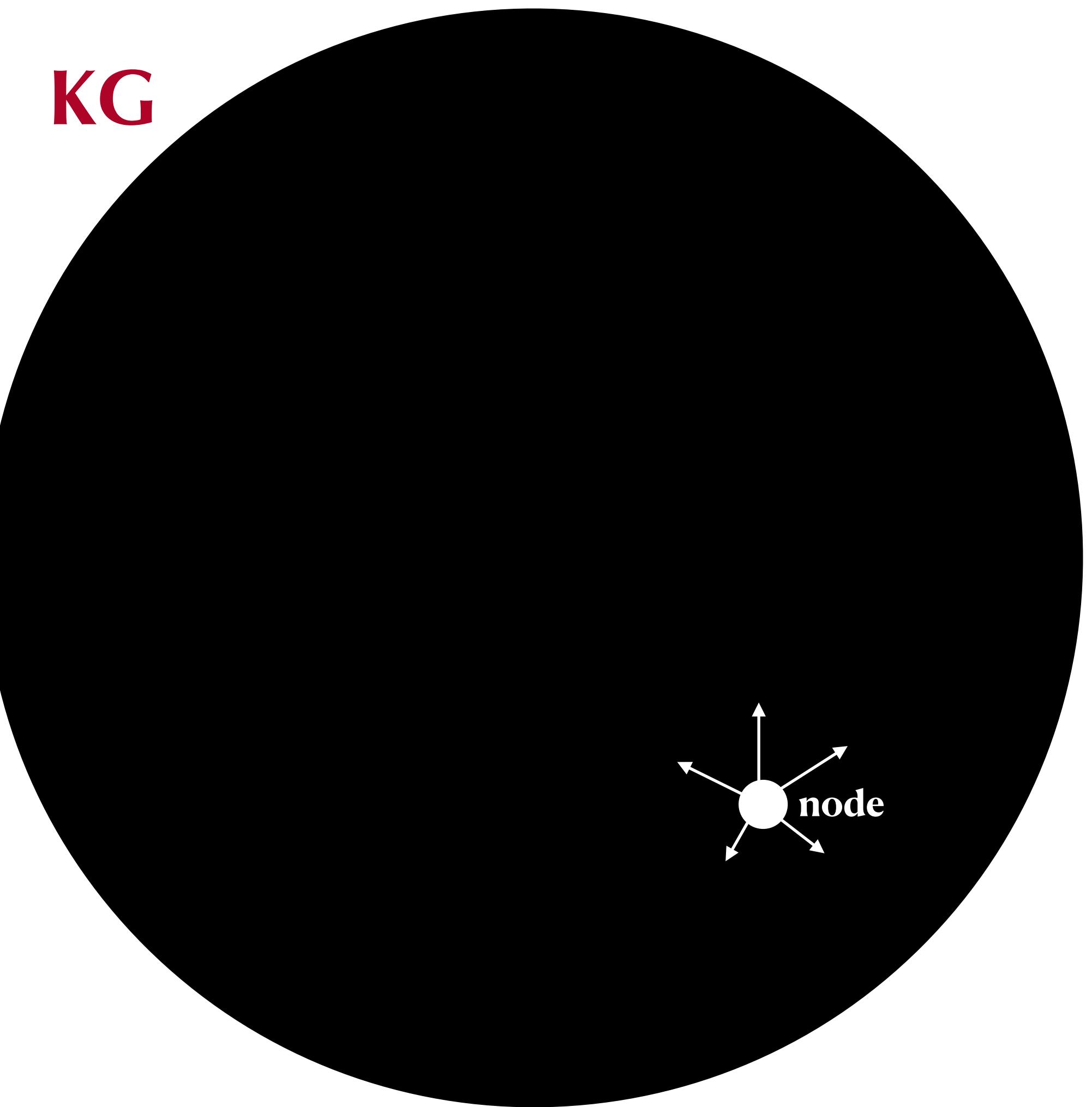
# Curiosity!



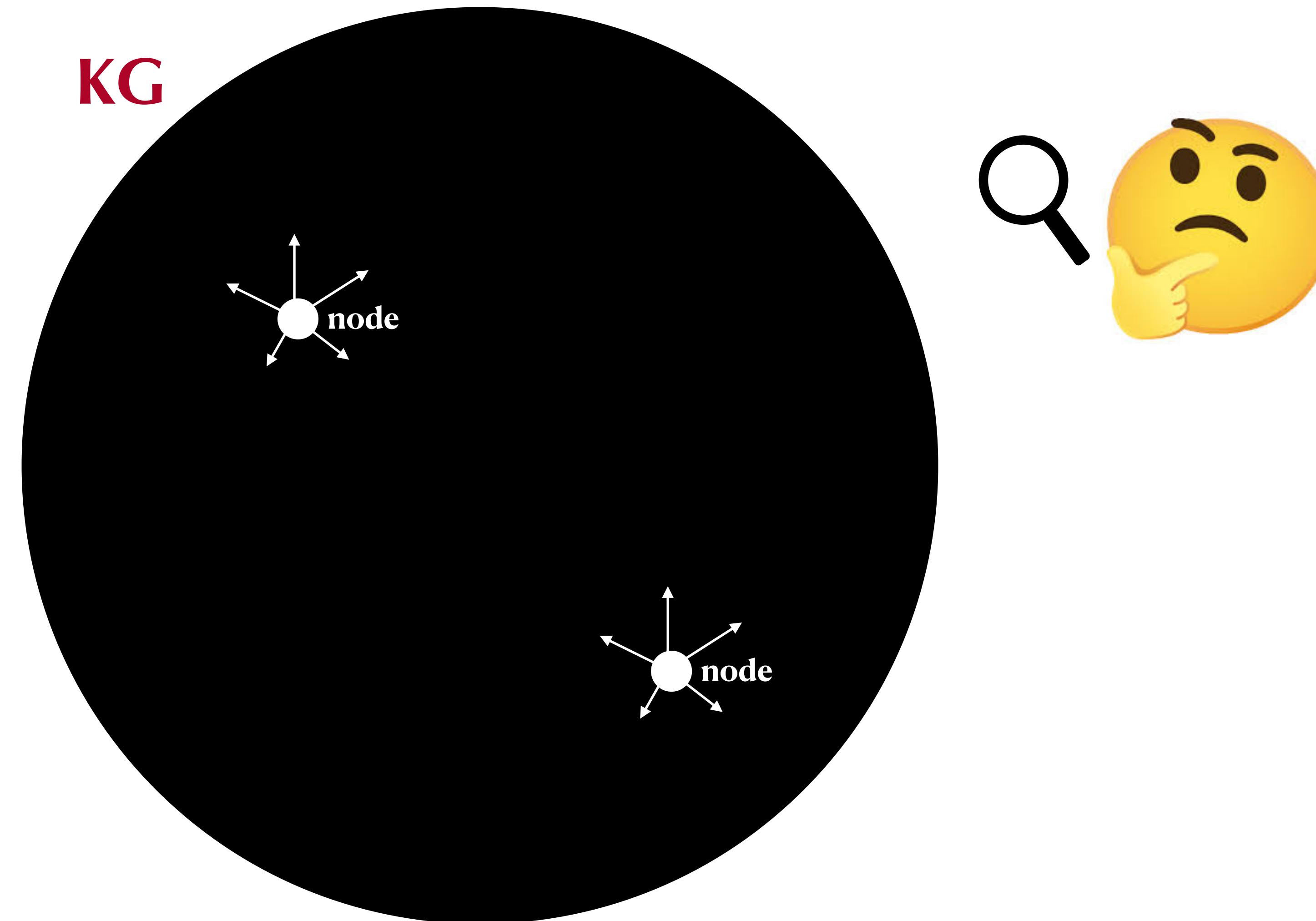
**KG**



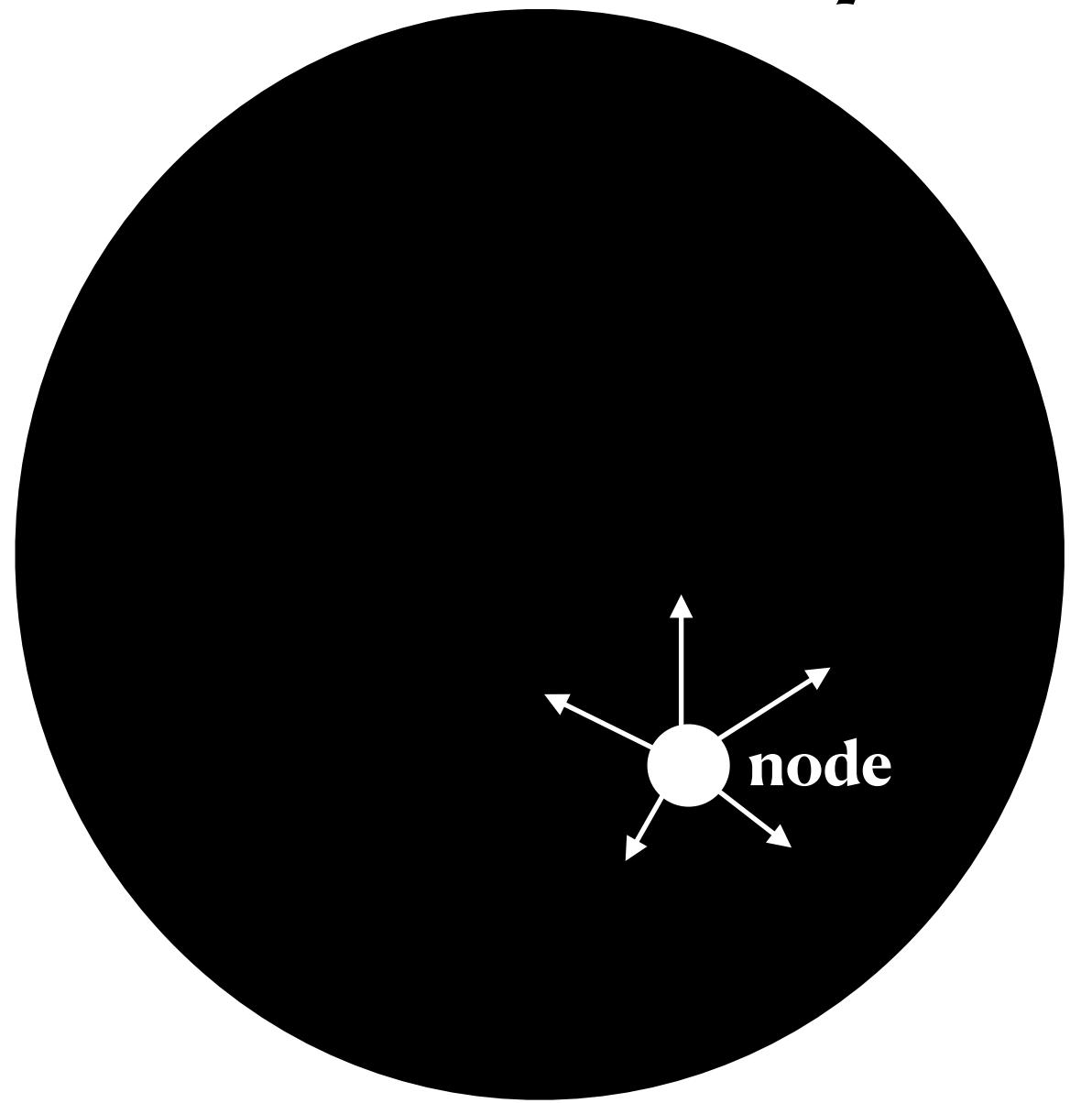
# Curiosity!



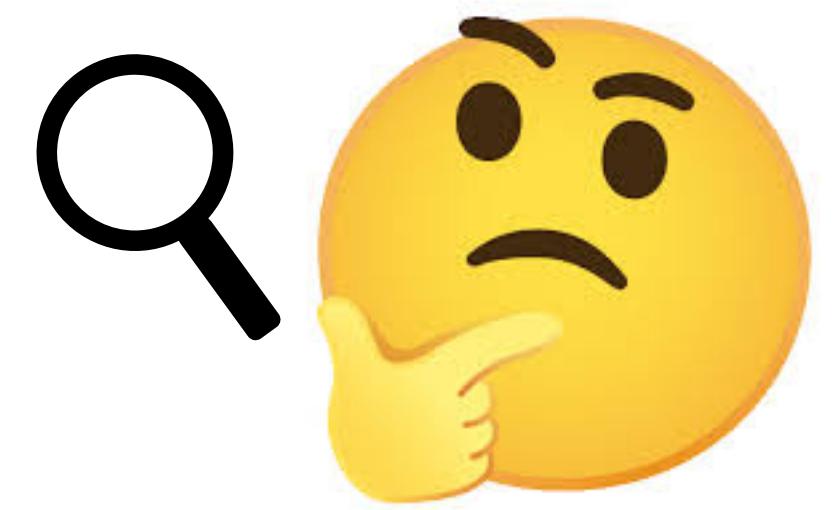
# Curiosity!



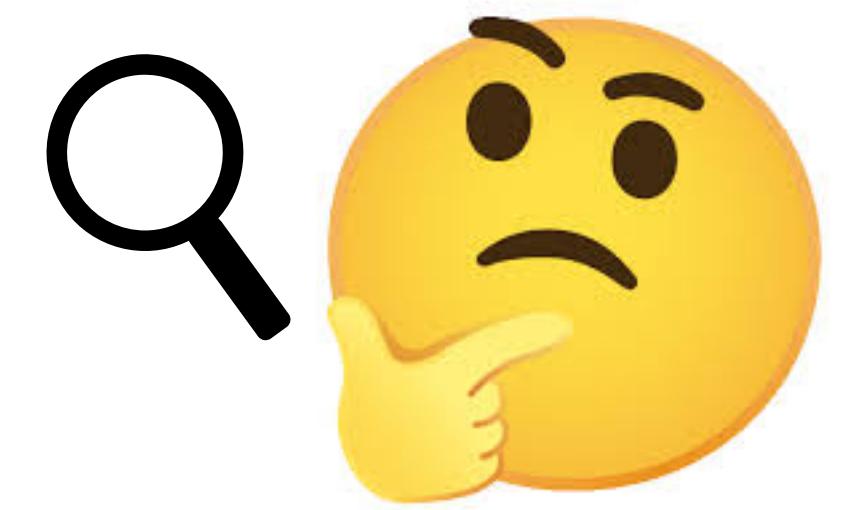
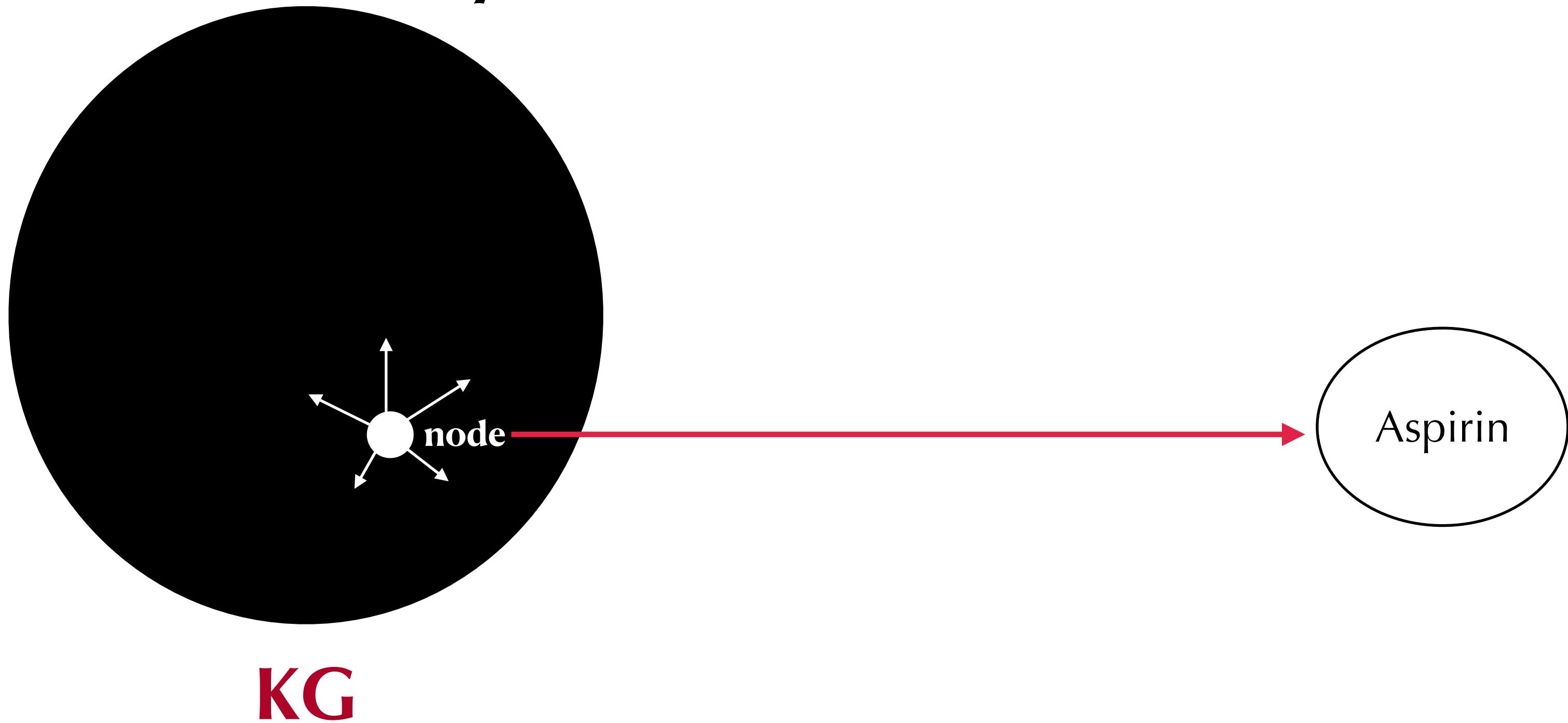
# Curiosity!



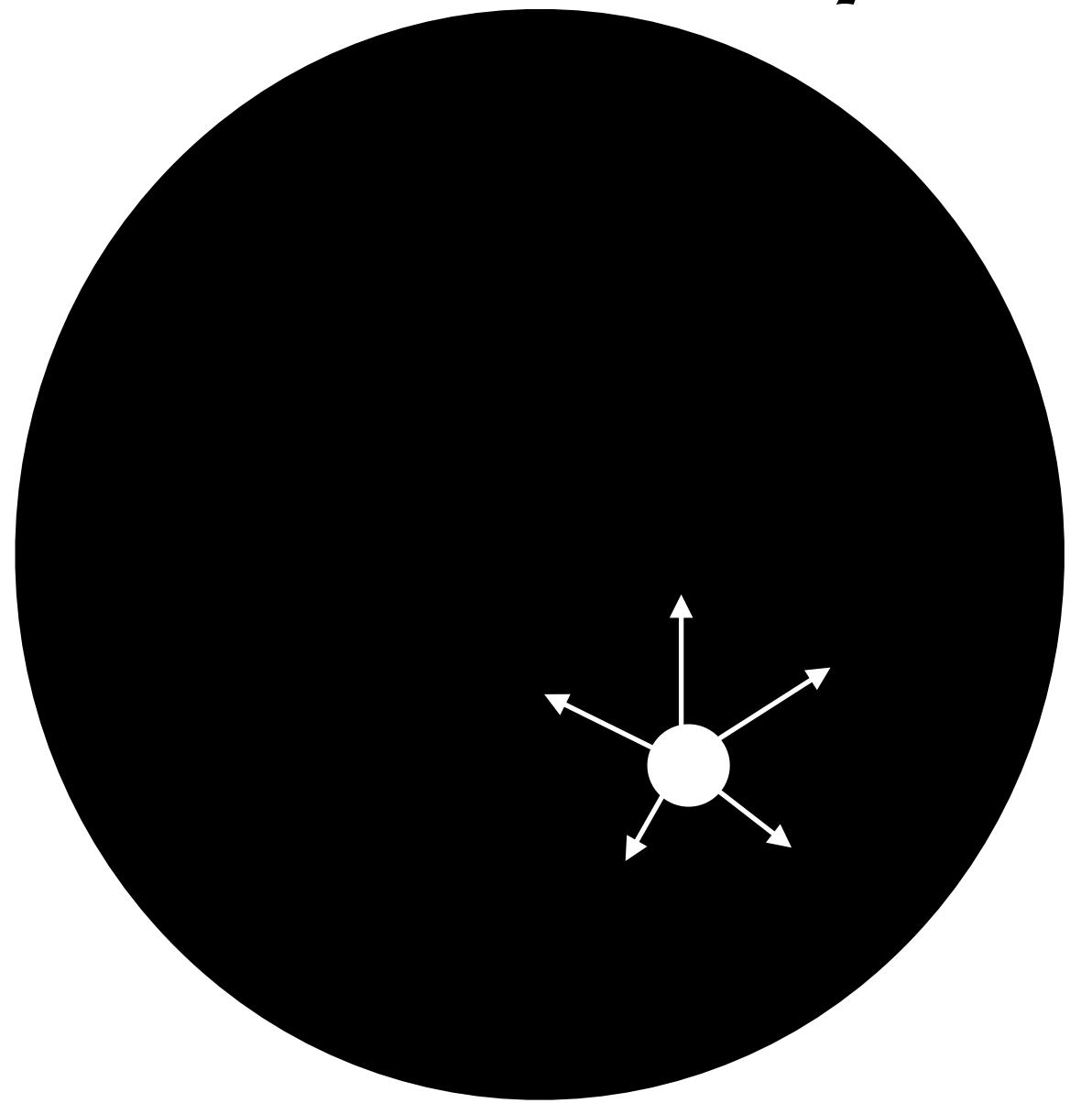
**KG**



# Curiosity!

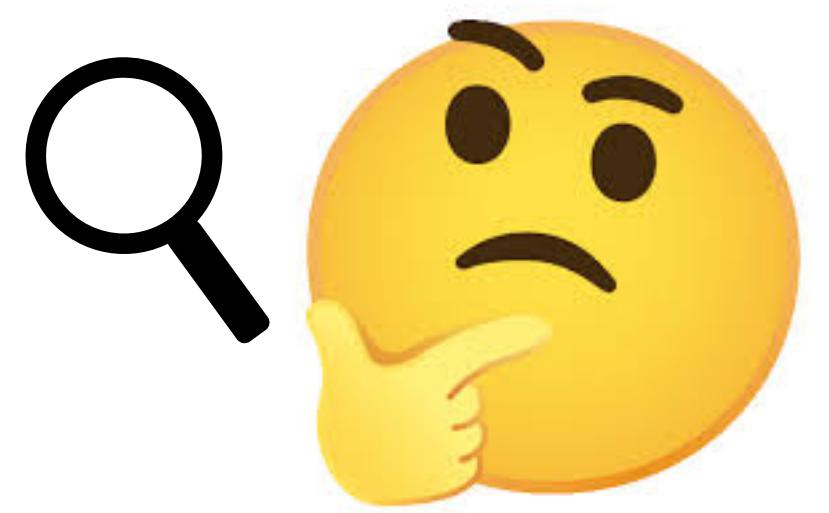


# Curiosity!

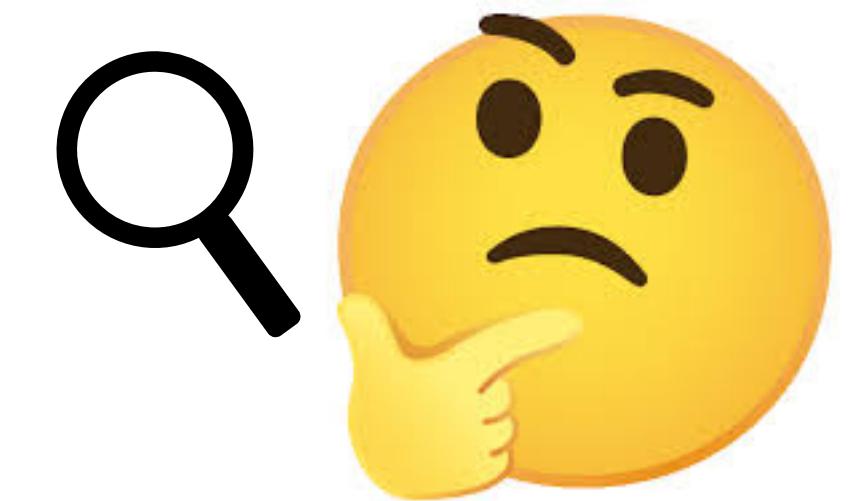
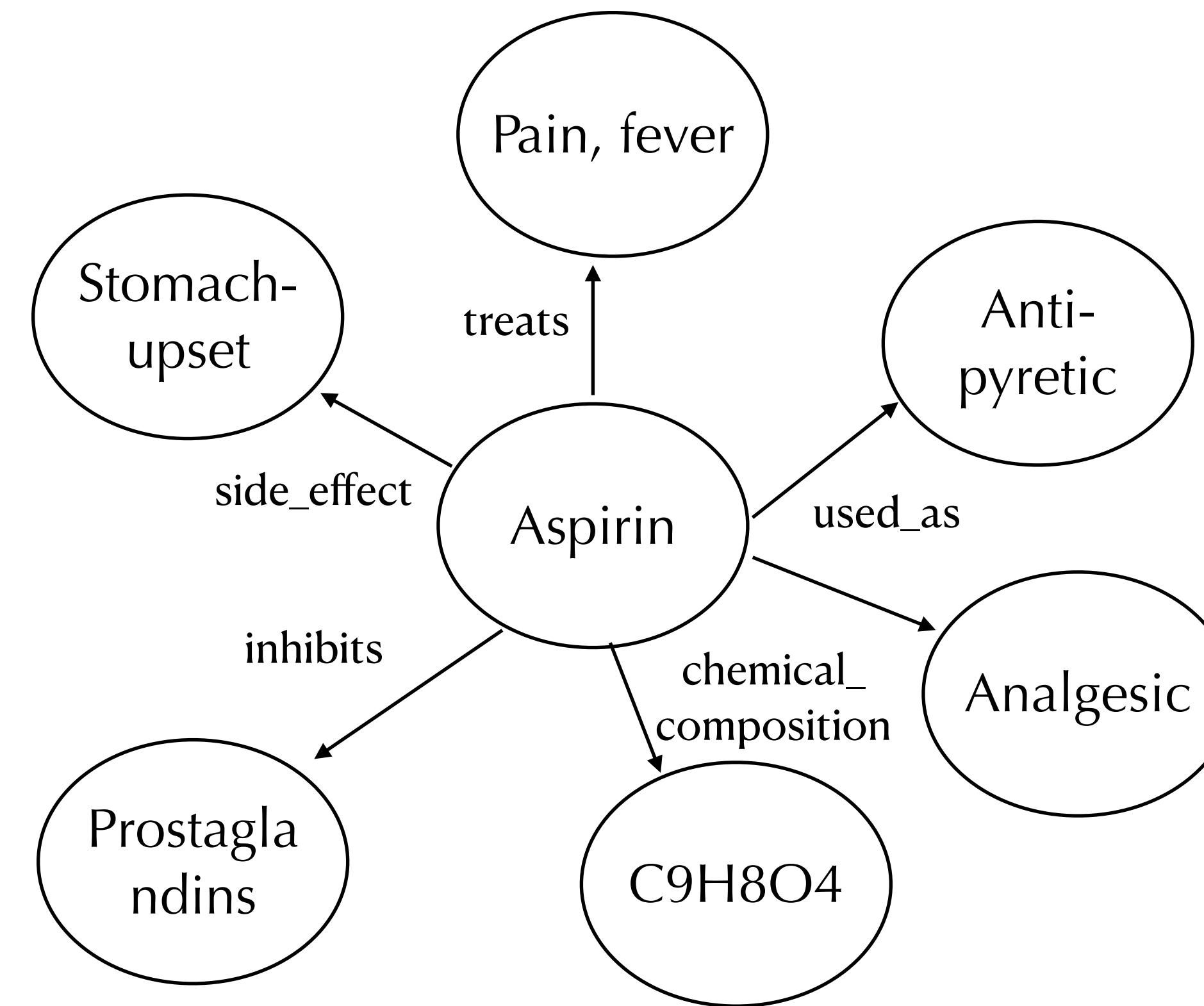
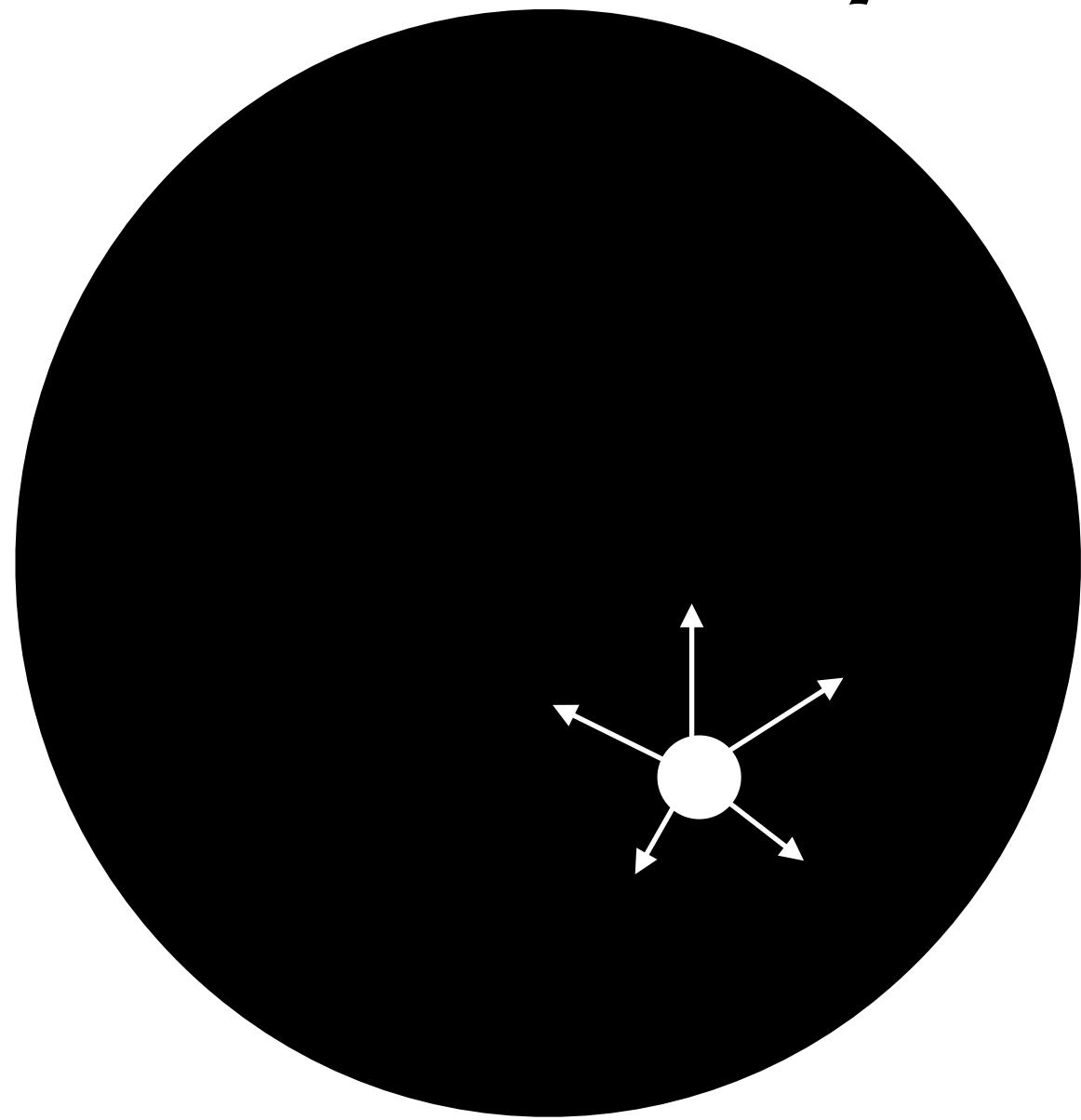


KG

Aspirin



# Curiosity!



# **Emerging Clarity**

# Emerging Clarity



# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements



# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements

The KG also has information about drugs, their side effects, their treatments, etc..



# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements

The KG also has information about drugs, their side effects, their treatments, etc..

**Therefore, the KG can answer questions like...**



# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements

The KG also has information about drugs, their side effects, their treatments, etc..

**Therefore, the KG can answer questions like...**



- ◆ Which year was Person X born?
- ◆ What is X's age?
- ◆ How many books X has written?
- ◆ Who is X married to?

# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements

The KG also has information about drugs, their side effects, their treatments, etc..

Therefore, the KG can answer questions like...



- ◆ Which year was Person X born?
- ◆ What is X's age?
- ◆ How many books X has written?
- ◆ Who is X married to?

- ◆ What is the chemical composition of Drug A?
- ◆ What are the side effects of A?
- ◆ How many distinct diseases can be treated by A?

# Emerging Clarity

The KG has information about famous people such as their place & date of birth, spouse and their professional achievements

The KG also has information about drugs, their side effects, their treatments, etc..

Therefore, the KG can answer questions like...



◆ Which year was Person X born?

◆ What is the chemical composition of Drug A?

**Can we mechanize this human behavior?**

◆ Who is X married to?

◆ How many distinct diseases can be treated by A?

# Outline: Rest of the talk

- ◆ Task Formalization
- ◆ BYOKG Approach
- ◆ Stage 1: Exploration
- ◆ Stage 2: Question Generation
- ◆ Stage 3: Reasoning
- ◆ Results
- ◆ Future directions

# Task: KGQA (Program Synthesis)

Given **KG**:  $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times (\mathcal{E} \cup \mathcal{L} \cup \mathcal{C})$

Find answer set  $\mathcal{A}_q$  for a natural language query  $q$   
by mapping  $q$  to a program  $p_q$

s.t.  $\text{eval}^{\mathcal{K}}(p_q) = \mathcal{A}_q$

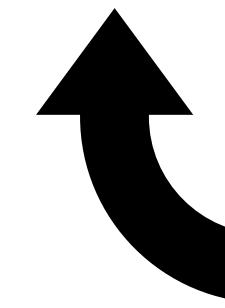
# Task: KGQA (Program Synthesis)

## Example:

$q$  Who are the sponsors of the Stanford Medicine X conference series?

$p_q$  (AND conferences.conference\_sponsor (JOIN  
conferences.conference\_sponsor.conferences  
m.0j2fyjs))

$\mathcal{A}_q$  {m.0c1d2\_9 (Stanford Anesthesia),  
m.02rkyb4 (Stanford Hospital & Clinics) }

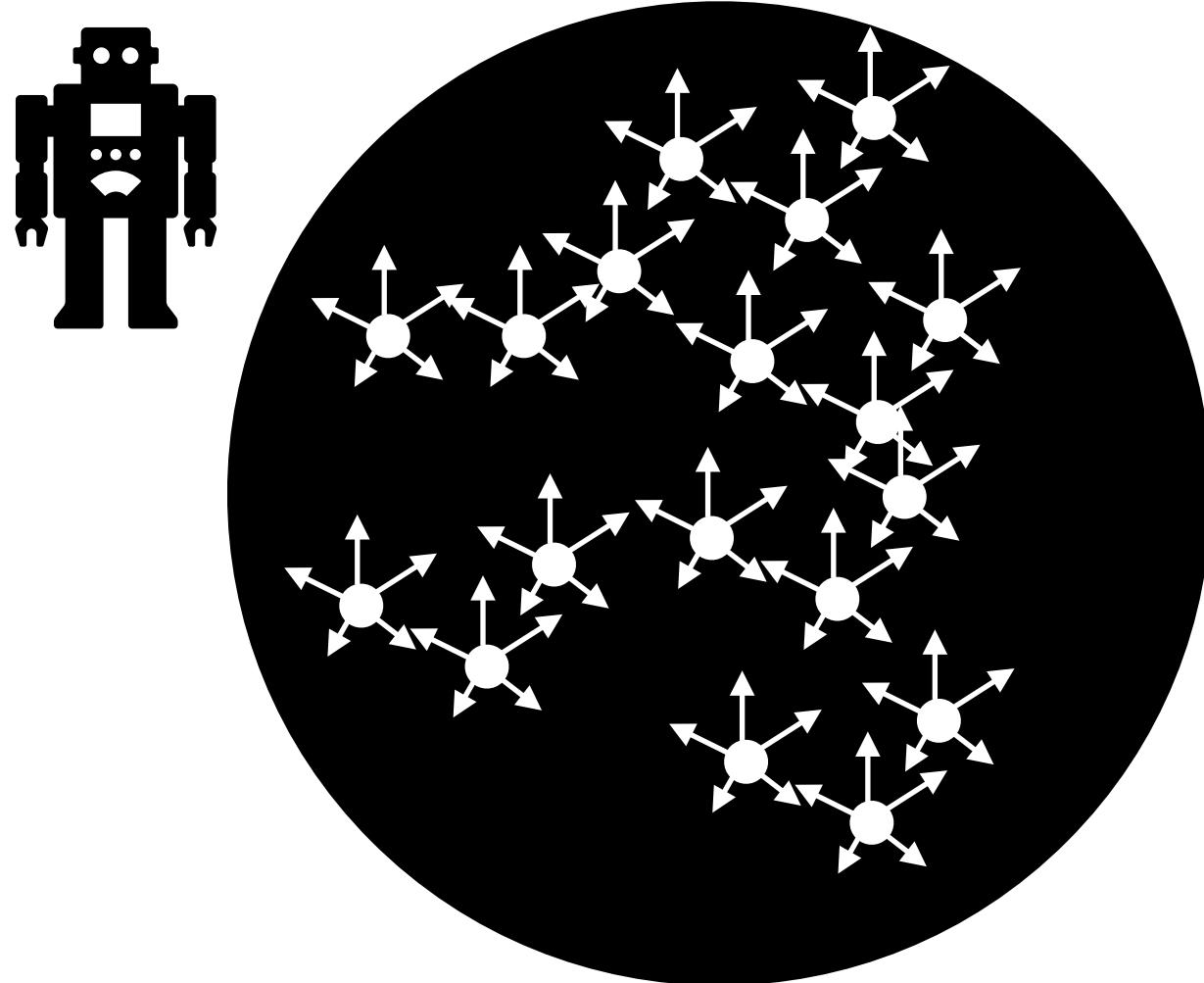


**S-expression**

# BYOKG: Approach

# BYOKG: Approach

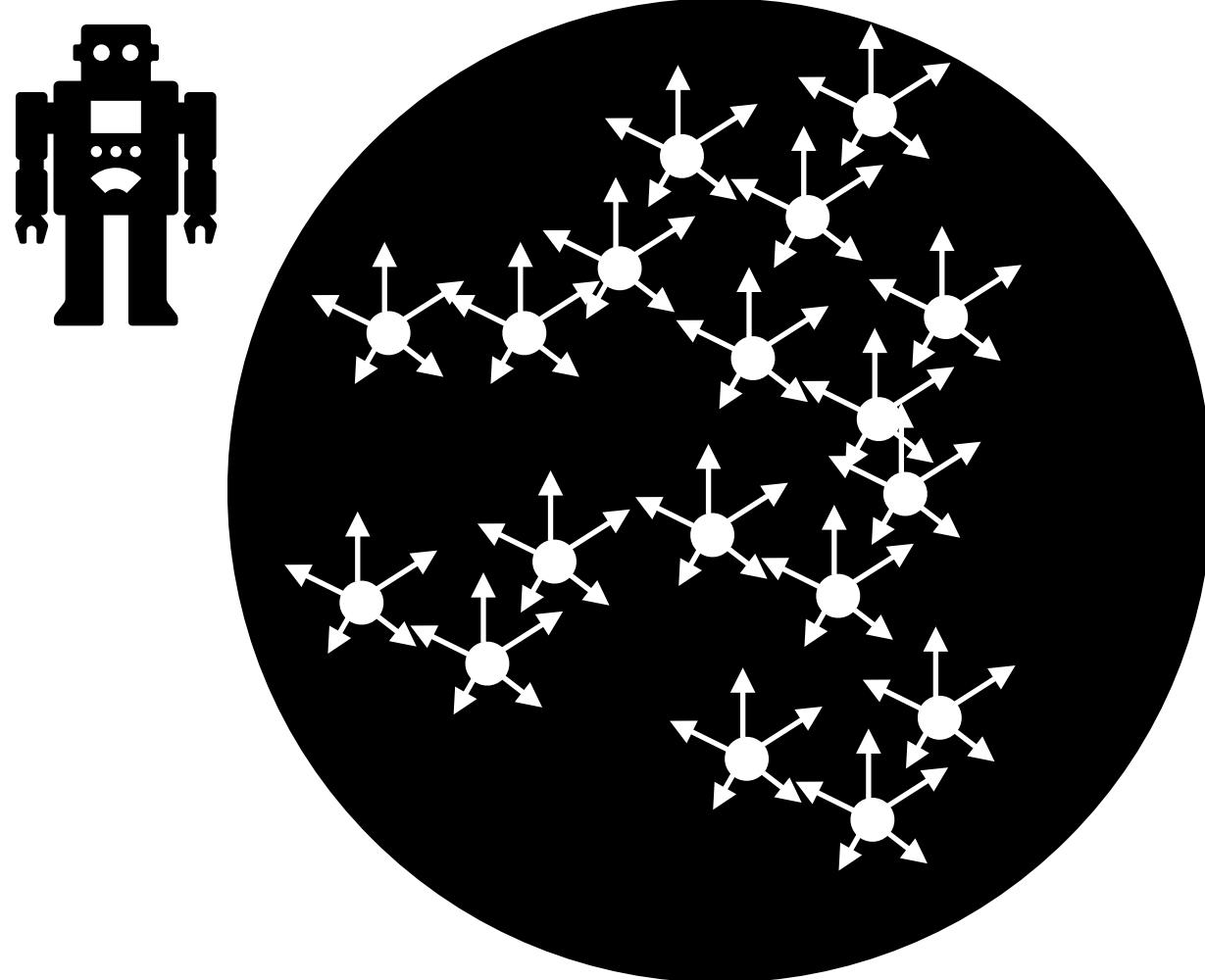
## Stage 1: Symbolic Graph Exploration



$$\mathcal{X}^P := \{p_i\}$$

# BYOKG: Approach

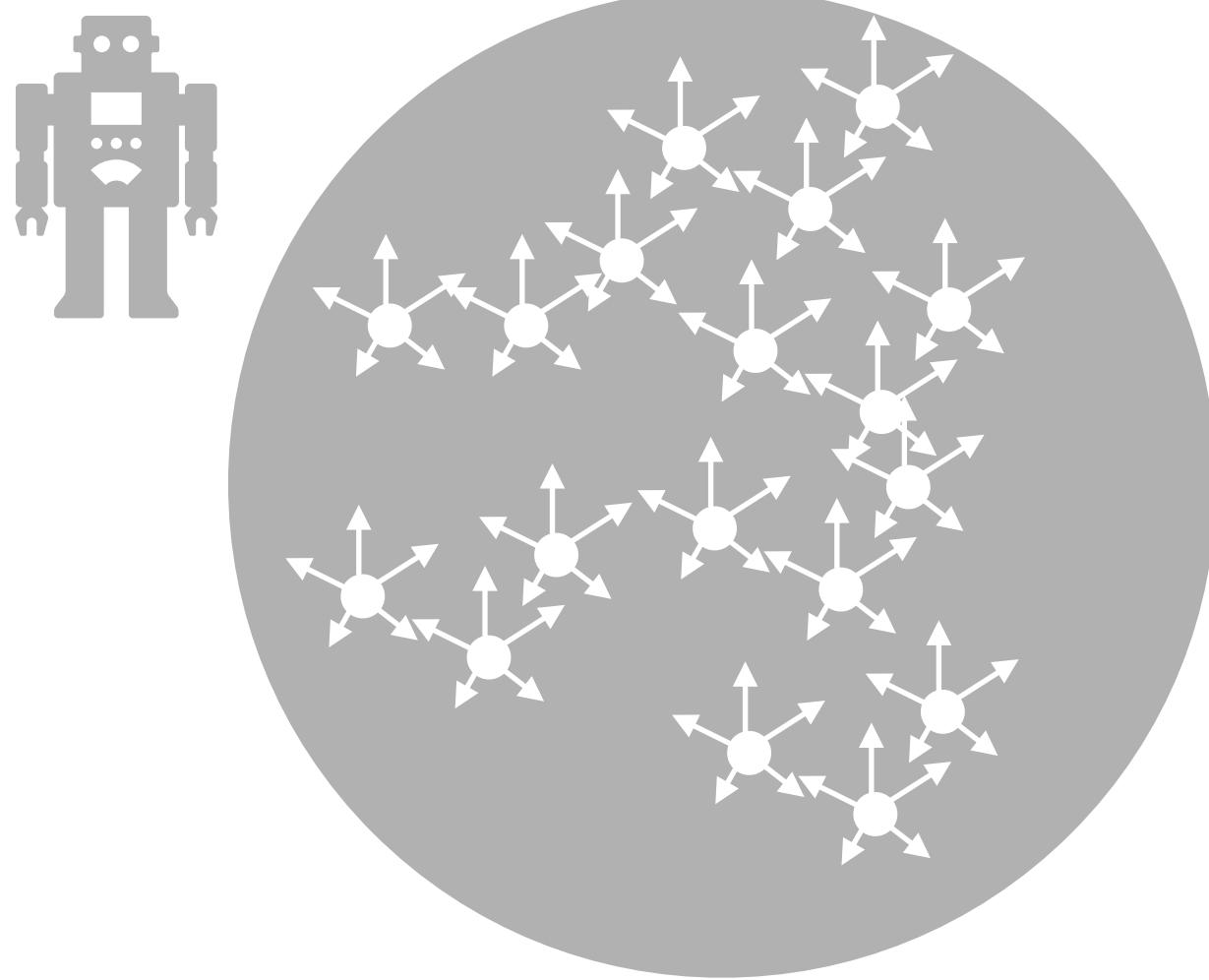
## Stage 1: Symbolic Graph Exploration



$$\mathcal{X}^P := \{p_i\}$$

# BYOKG: Approach

**Stage 1:**  
**Symbolic Graph Exploration**



$$\mathcal{X}^P := \{p_i\}$$

**Stage 2:**  
**NL Query Generation**

(JOIN radio.radio...m.01mxc7) )

**LLM:** What is the...?

(COUNT (AND computer.des...) )

**LLM:** How many...?

...

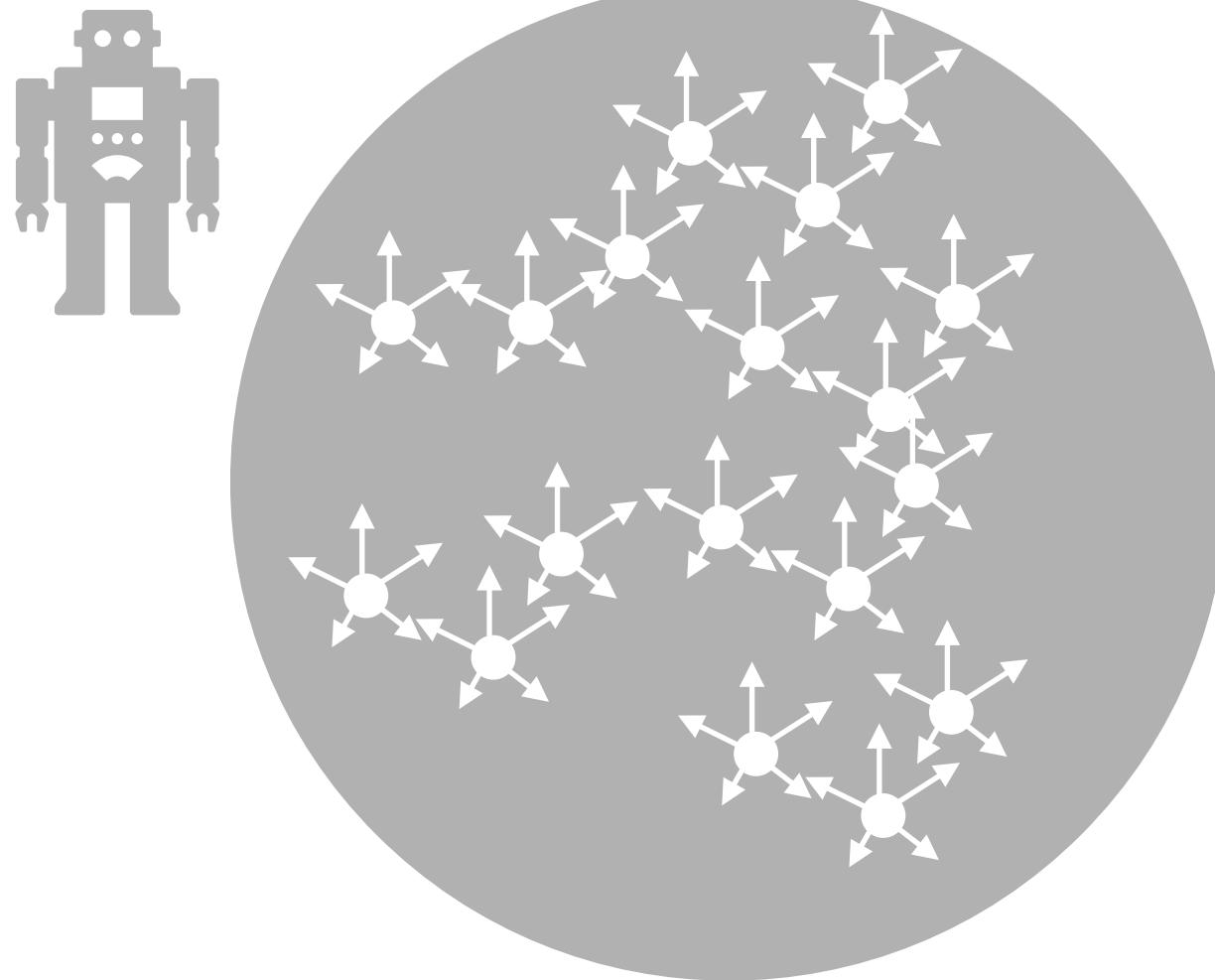
(ARGMAX (AND film.dire...) )

**LLM:** Who is the most...?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

# BYOKG: Approach

**Stage 1:**  
**Symbolic Graph Exploration**



$$\mathcal{X}^P := \{p_i\}$$

**Stage 2:**  
**NL Query Generation**

1. L2M prompting
2. Inverse-Consistency

(COUNT (AND computer.des...))

**LLM:** How many...?

...

(ARGMAX (AND film.dire...))

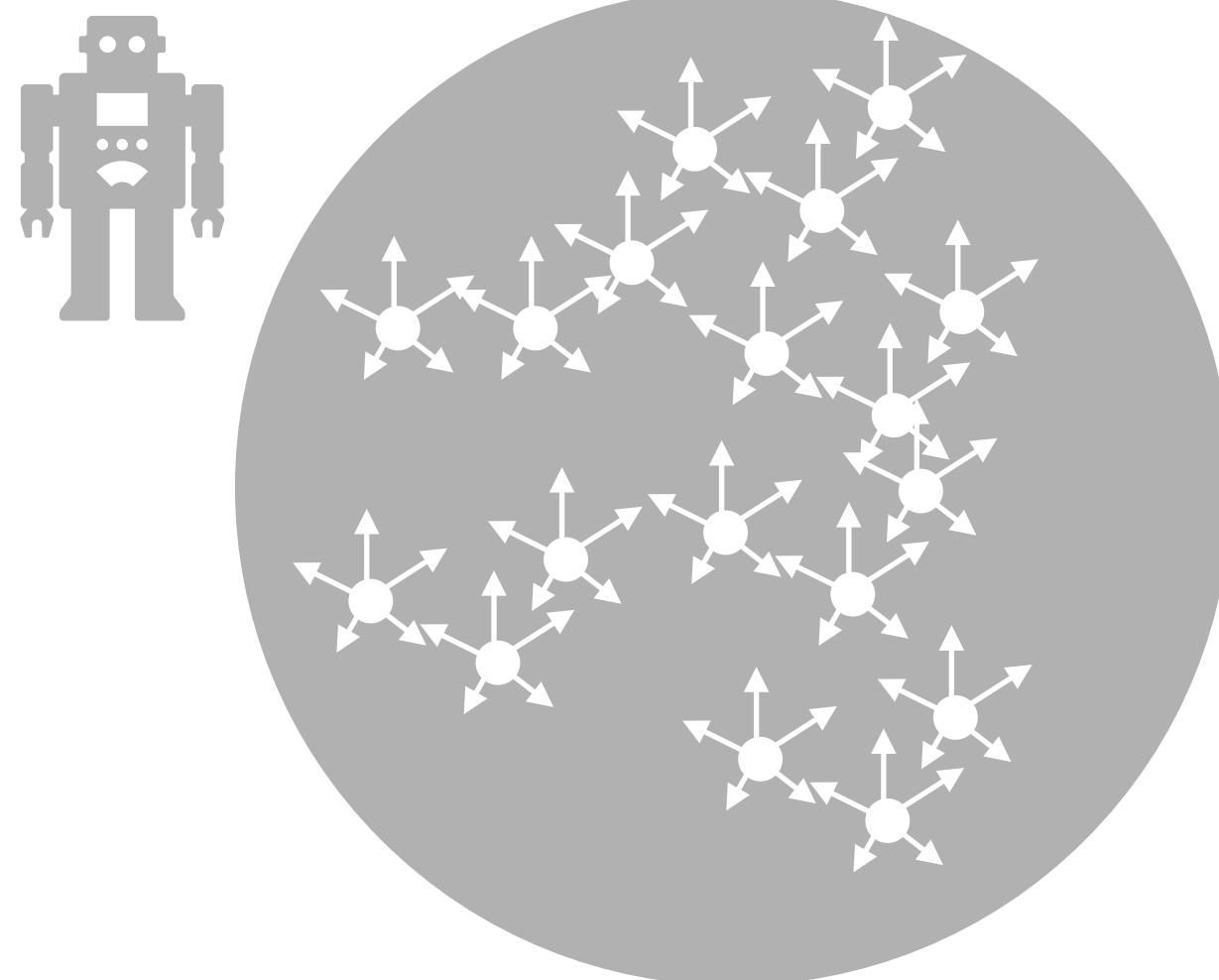
**LLM:** Who is the most...?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

# BYOKG: Approach

Required to be done once for a KG

## Stage 1: Symbolic Graph Exploration



$$\mathcal{X}^P := \{p_i\}$$

## Stage 2: NL Query Generation

1. L2M prompting
2. Inverse-Consistency

(JO COUNT (AND computer.des...))  
LL ...

**LLM:** How many...?

...

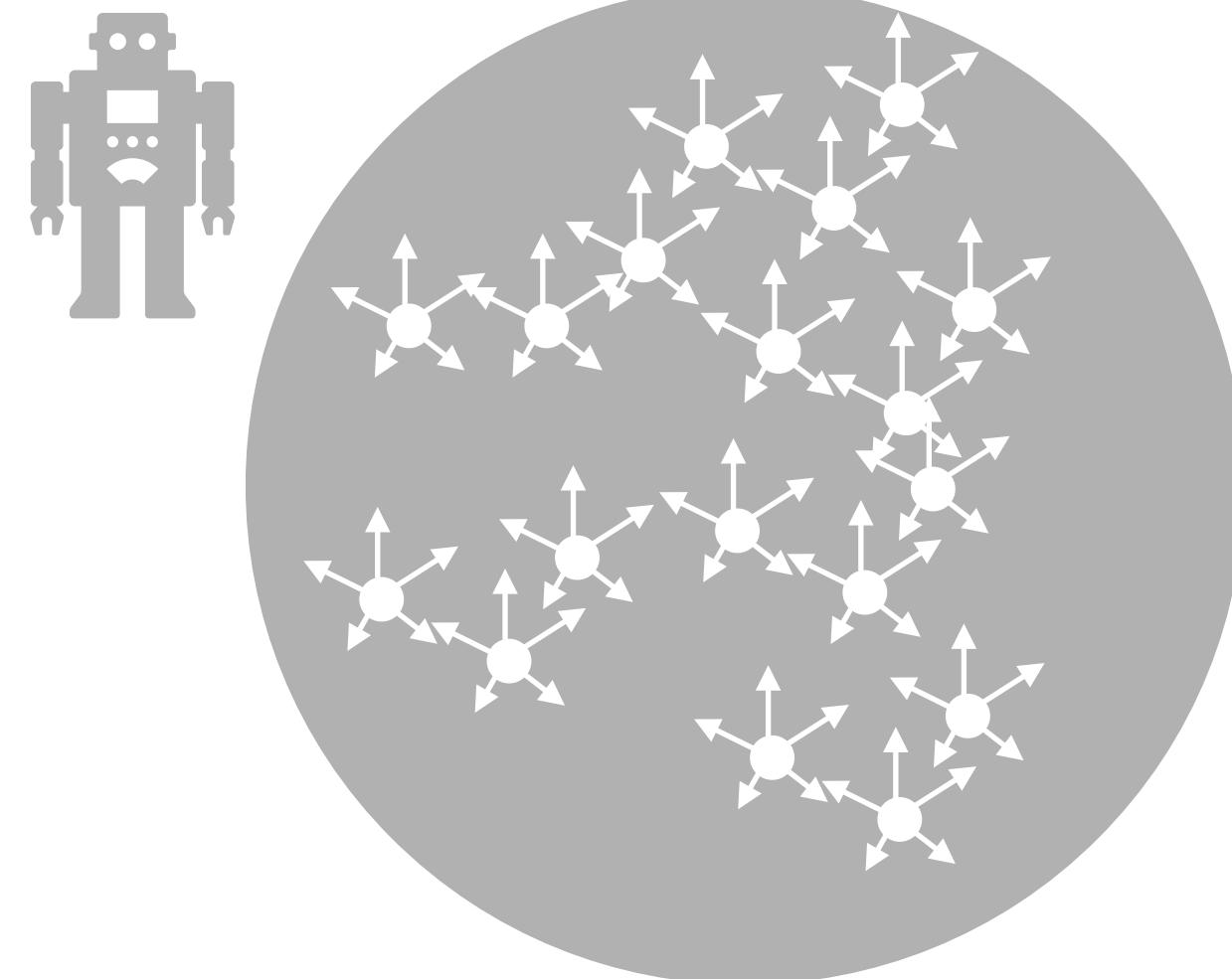
(ARGMAX (AND film.dire...))

**LLM:** Who is the most...?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

# BYOKG: Approach

## Stage 1: Symbolic Graph Exploration



$$\mathcal{X}^P := \{p_i\}$$

## Stage 2: NL Query Generation

1. L2M prompting  
(JOIN <sub>LLM</sub> ... )  
2. Inverse-Consistency

(COUNT (AND computer.des...))

LLM: How many...?

...

(ARGMAX (AND film.dire...))

LLM: Who is the most...?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

## Stage 3: Bottom-up Reasoning

Q: What is the name of the tallest art director whose profession is director of photography?



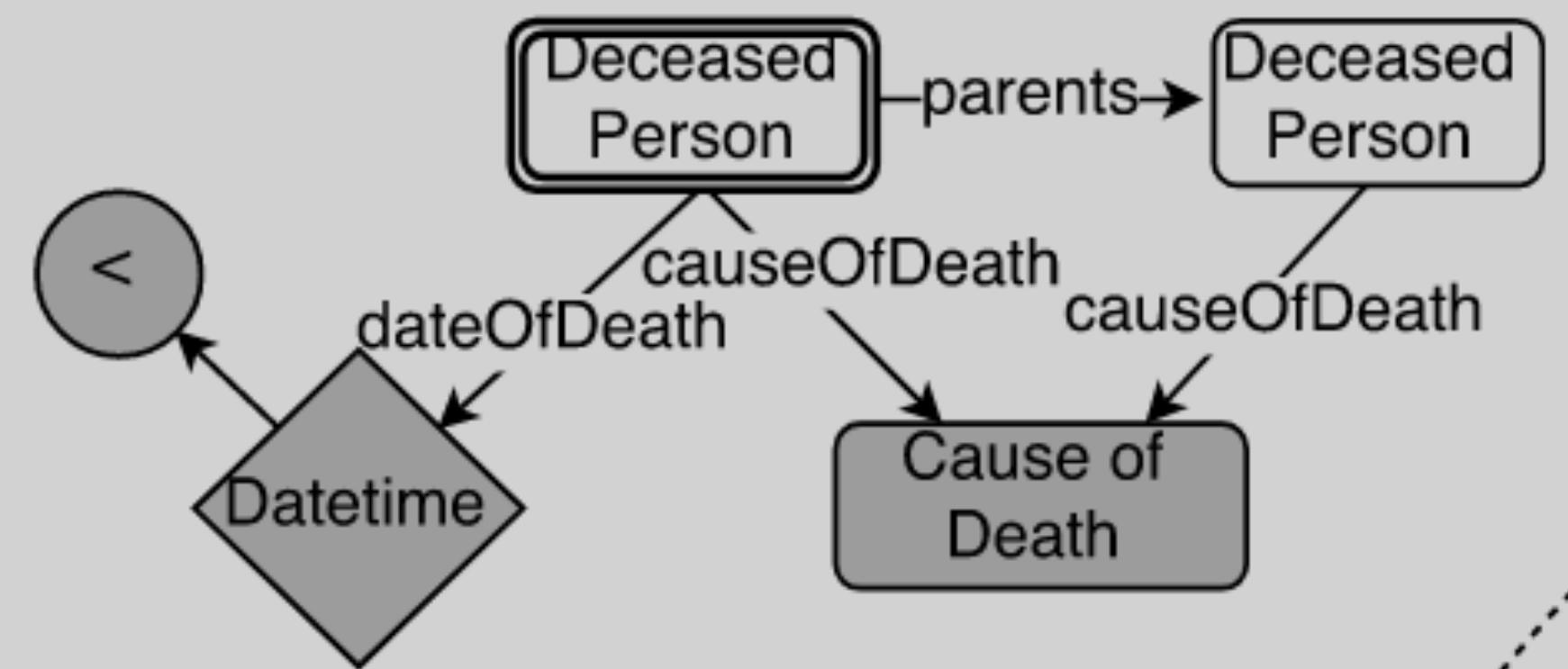
**p1:** m.0dgd\_

**p2:** (JOIN profession **p1**)

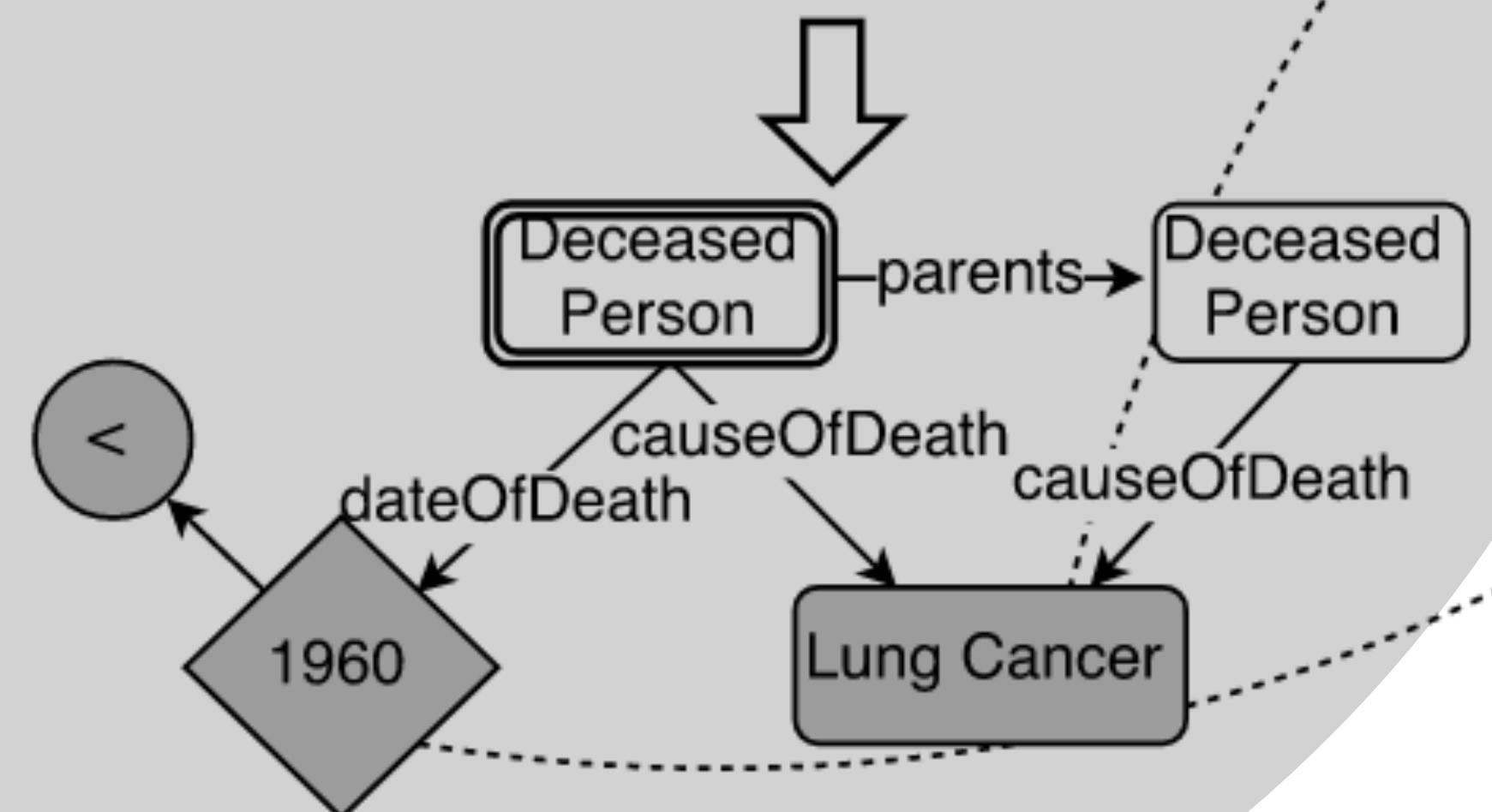
**p3:** (AND art\_director **p2**)

**p4:** (ARGMAX **p3** person.height)

# Stage 1: Symbolic Graph Exploration

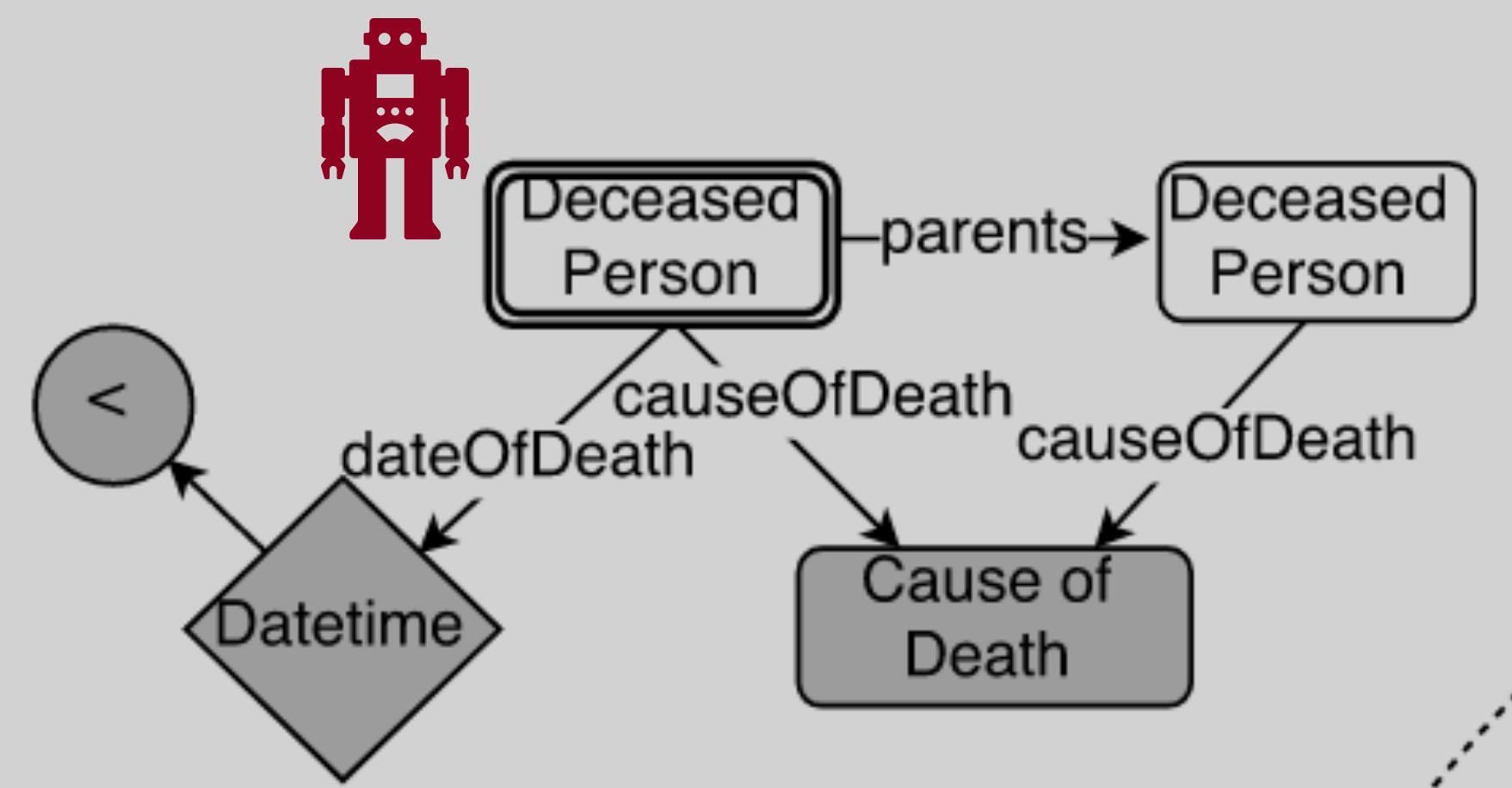


(b) Query template

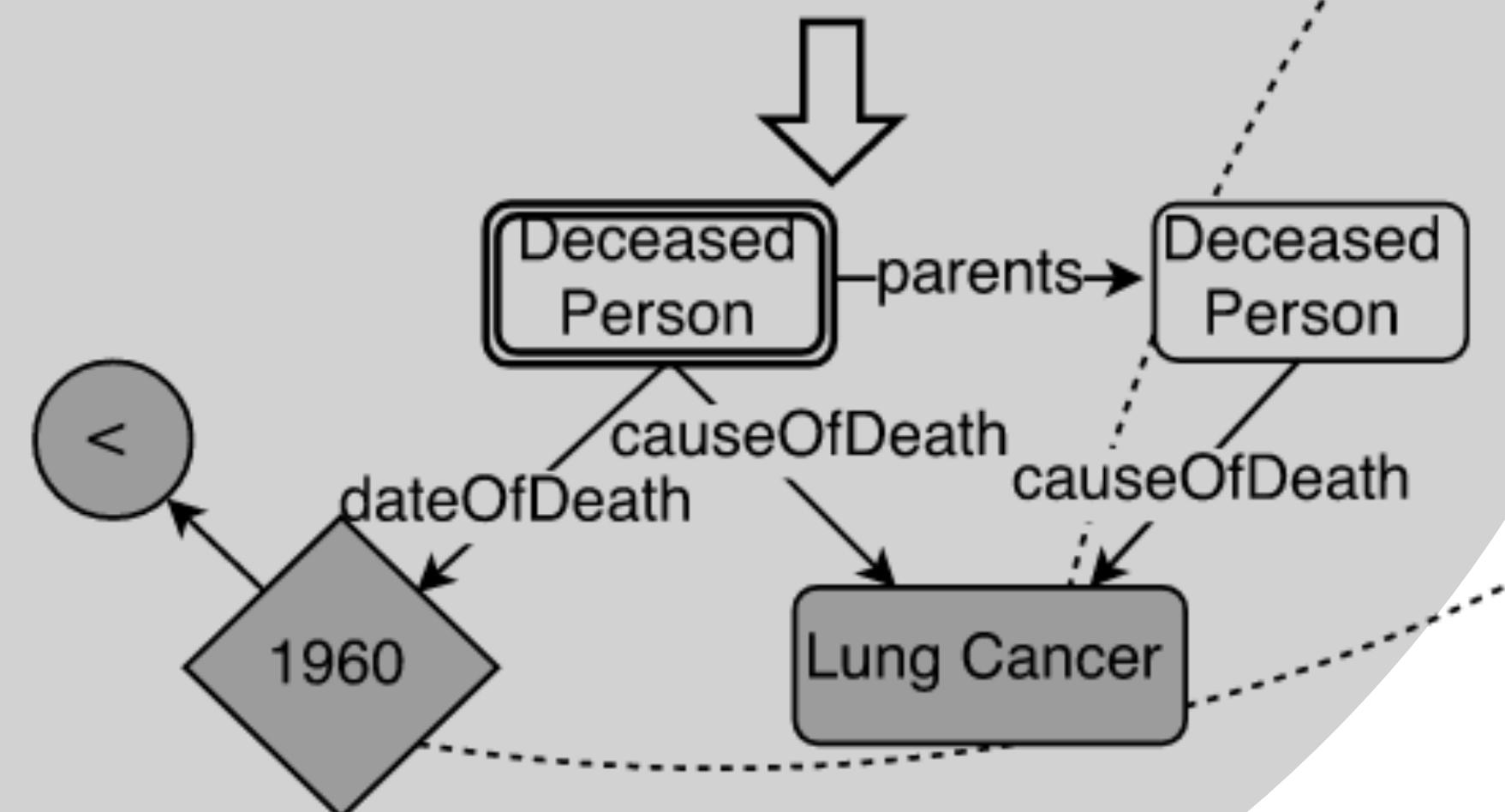


(c) Graph query

# Stage 1: Symbolic Graph Exploration



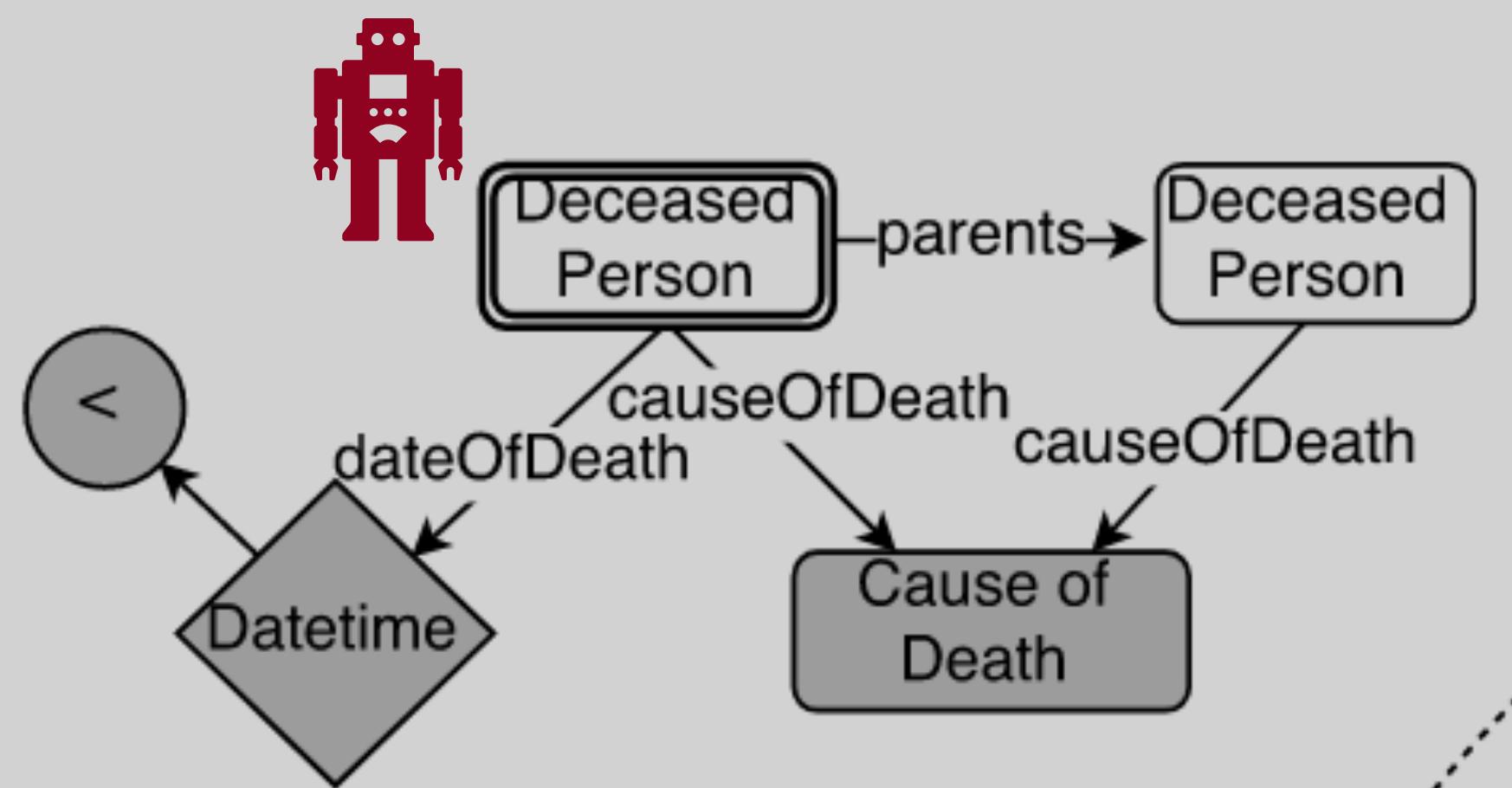
(b) Query template



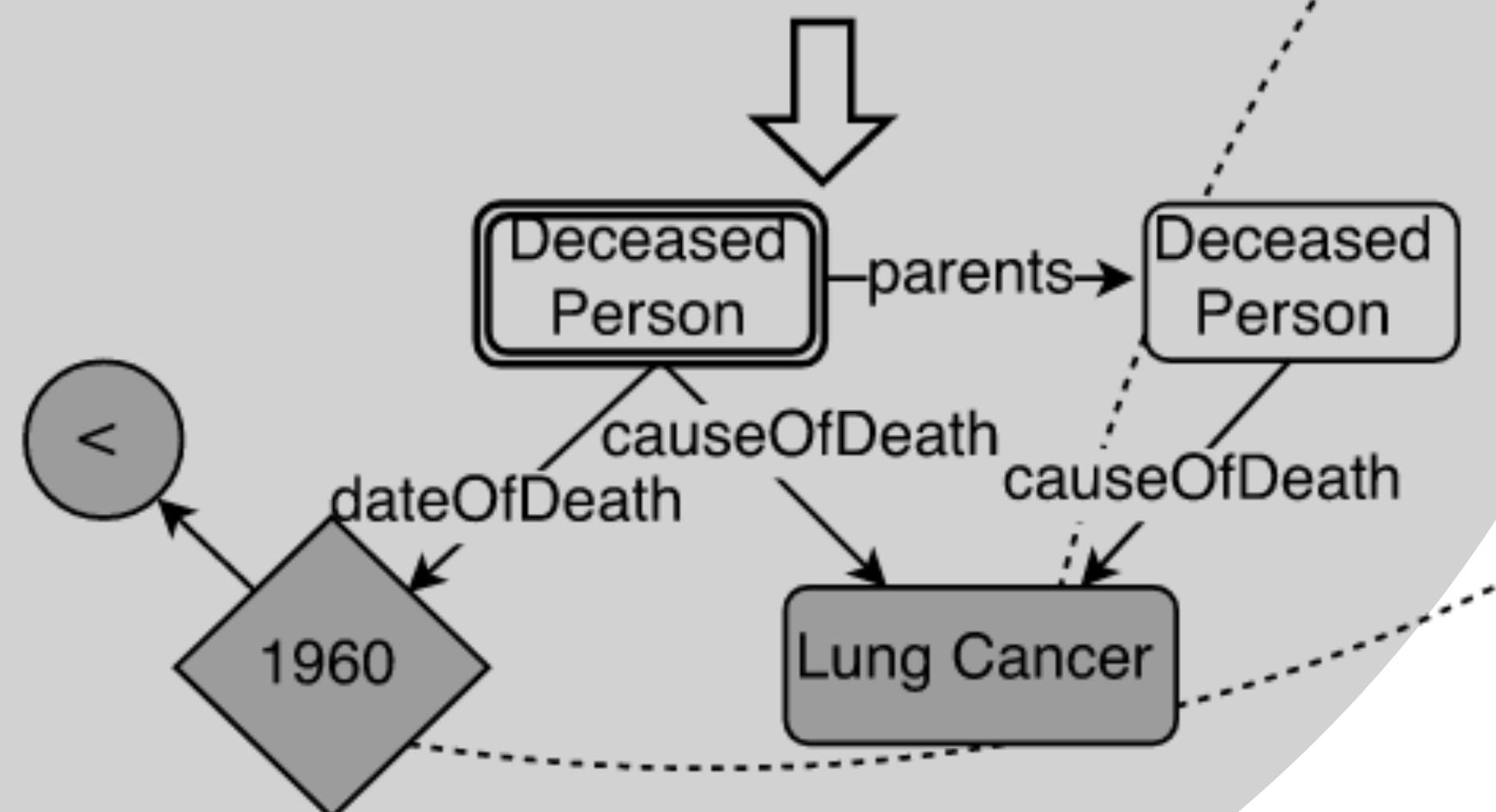
(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

# Stage 1: Symbolic Graph Exploration



(b) Query template

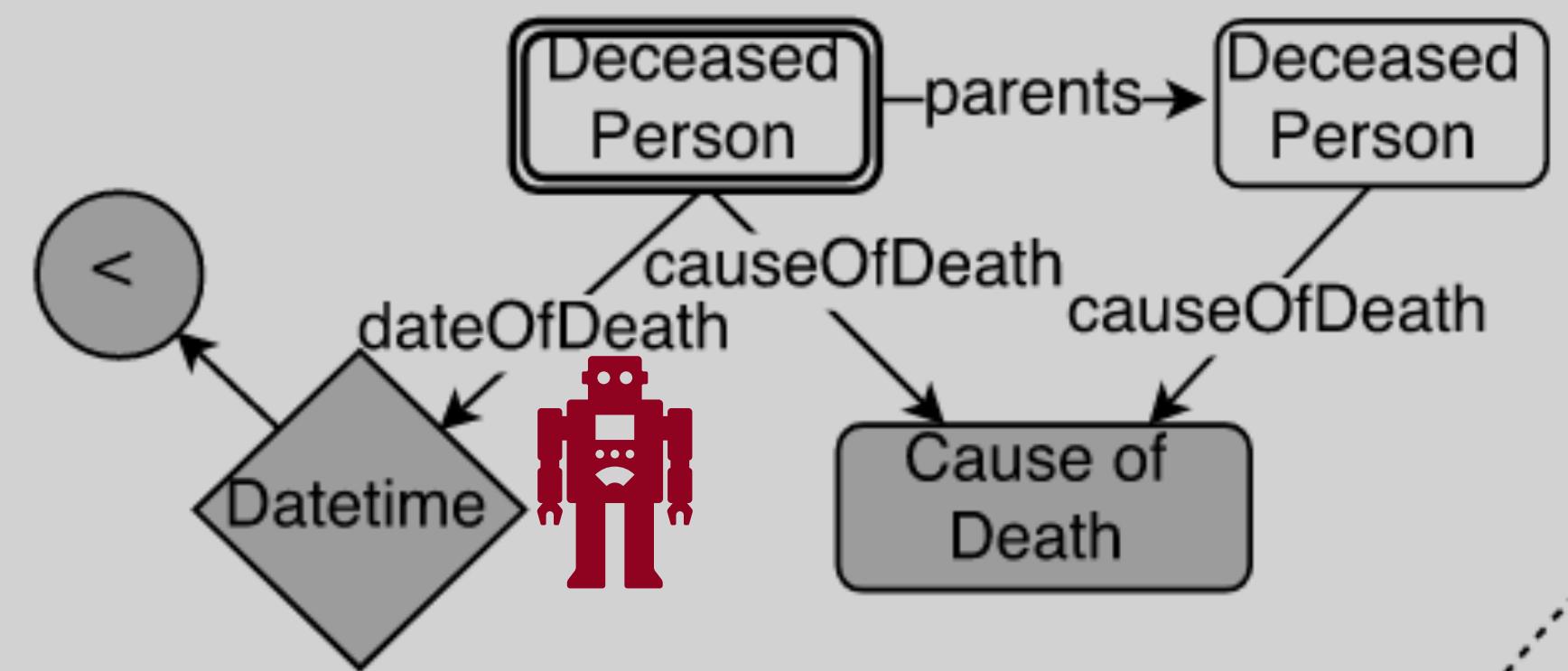


(c) Graph query

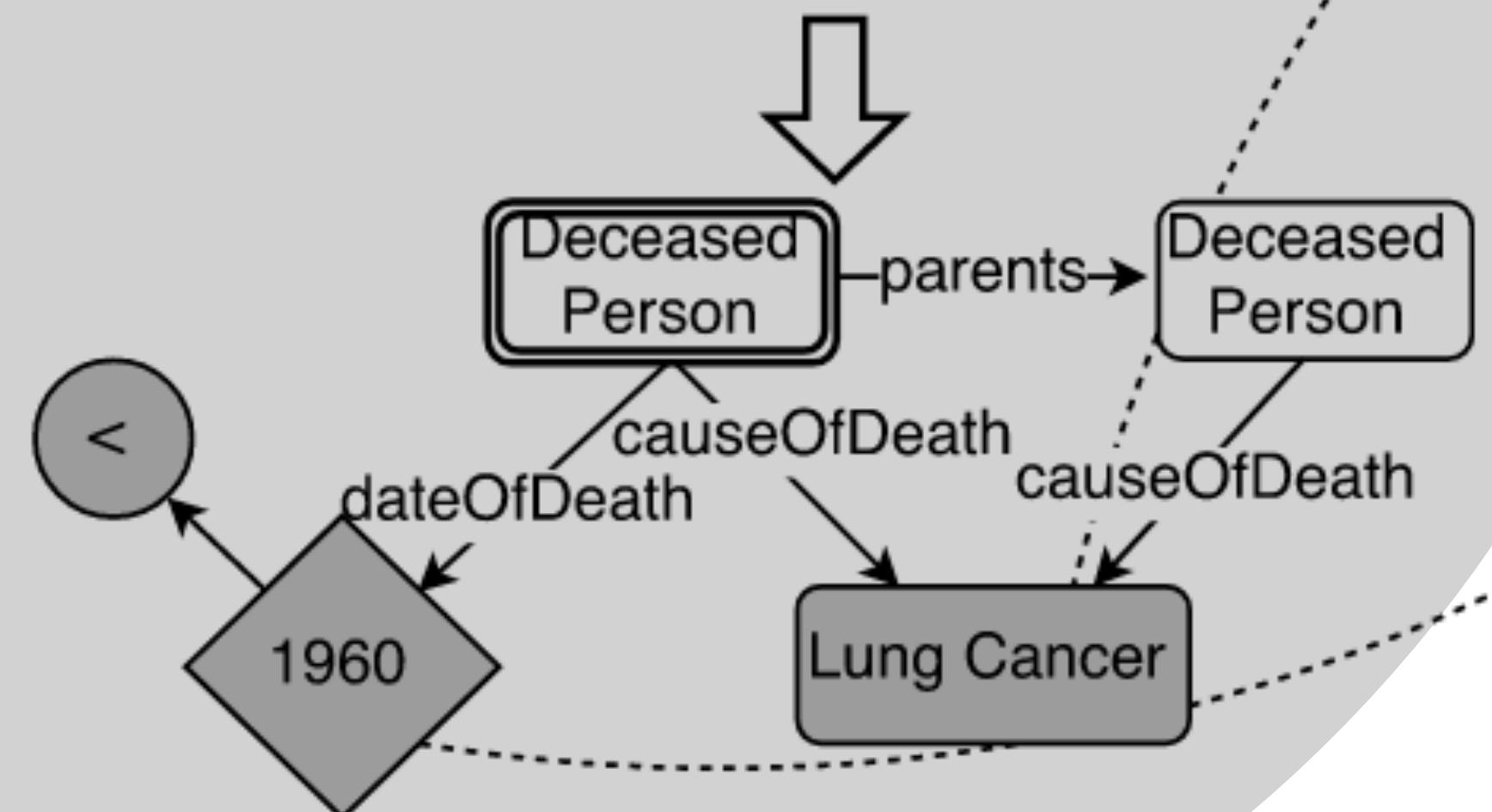
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

# Stage 1: Symbolic Graph Exploration



(b) Query template



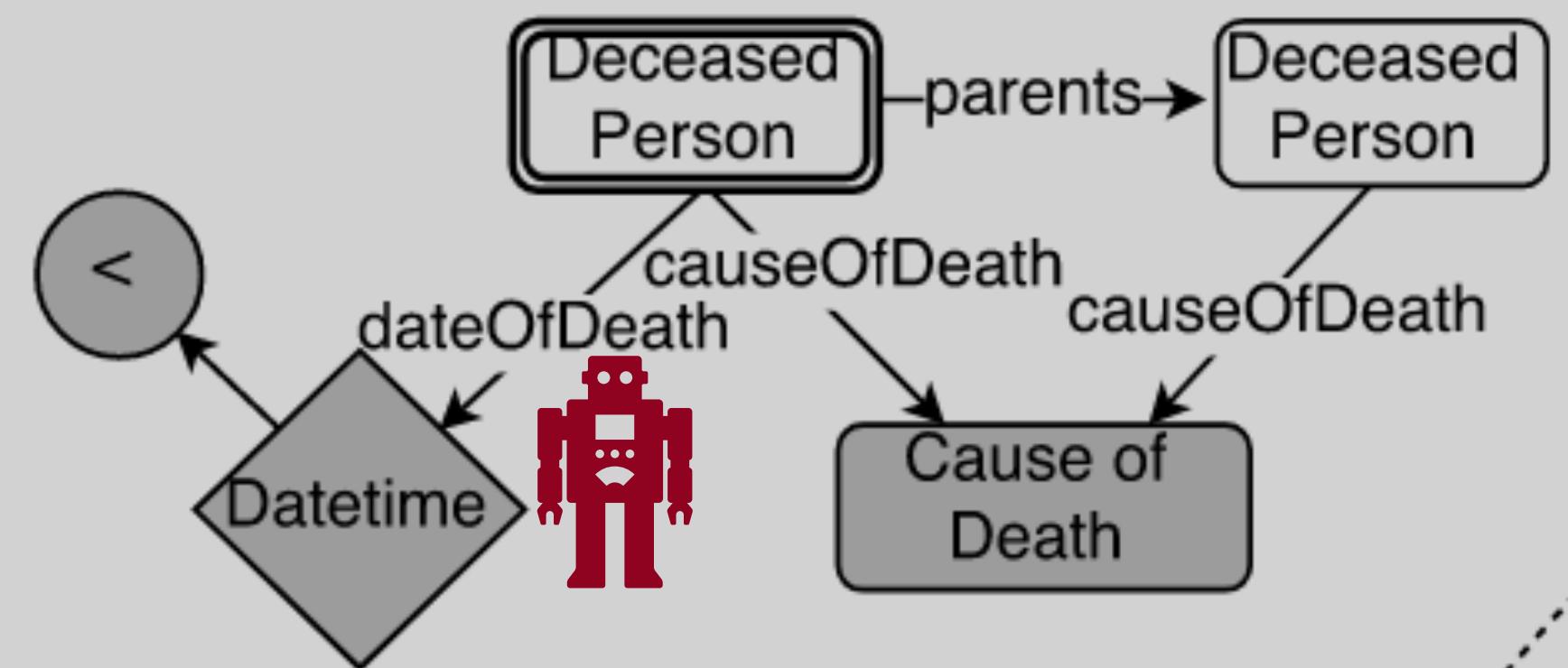
(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

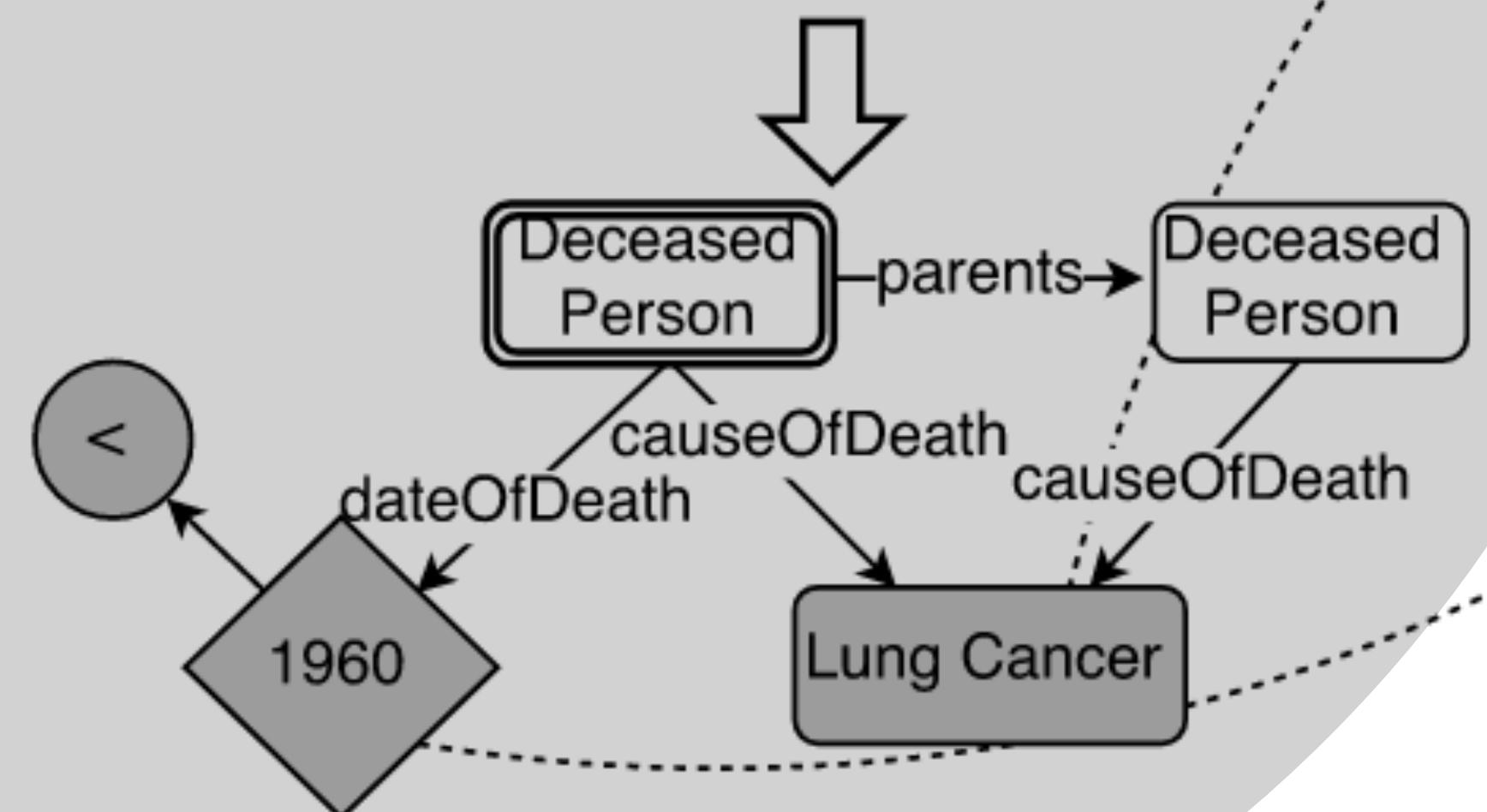
$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

$$p_1 := \text{extend}(p_0, s_0)$$

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

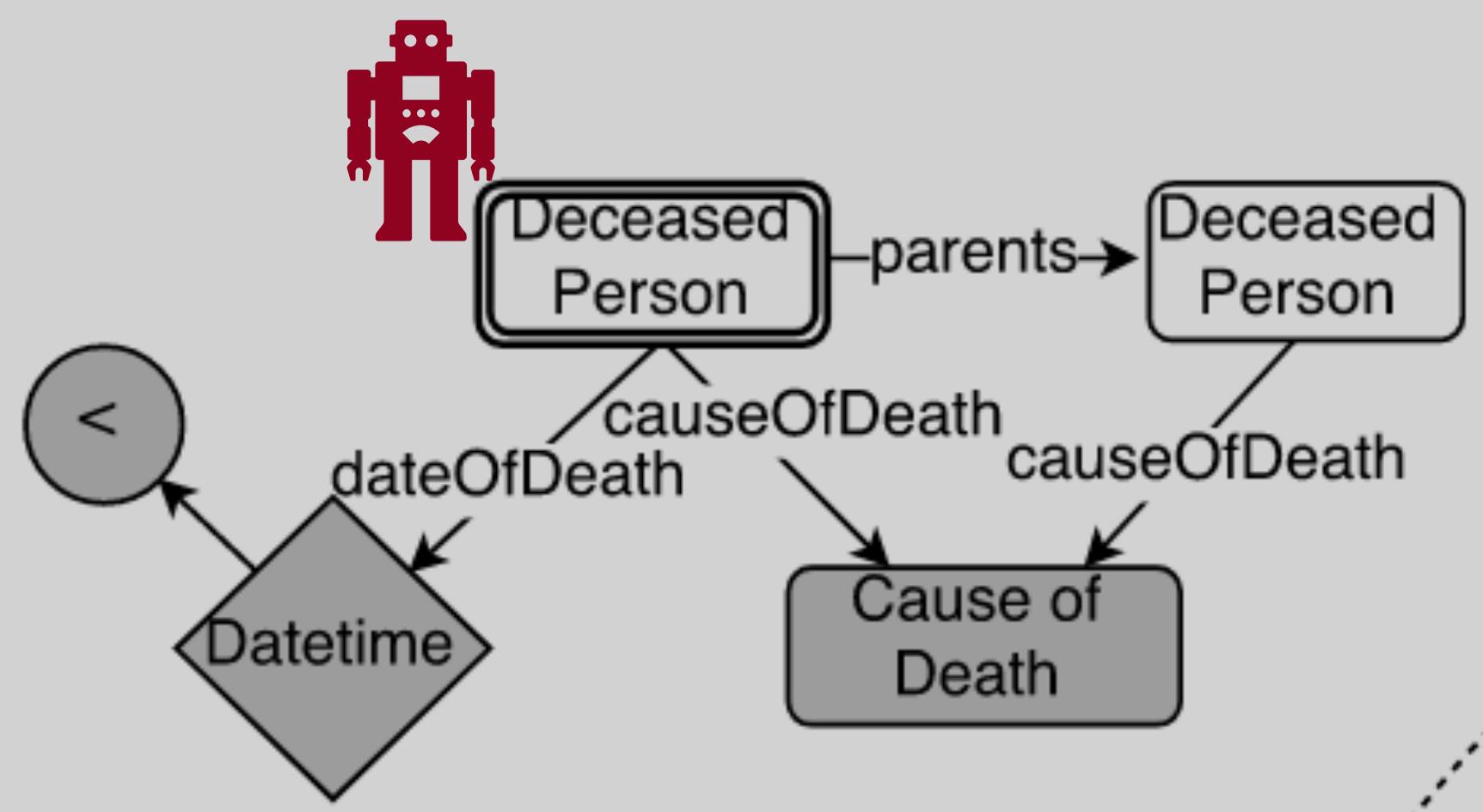
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

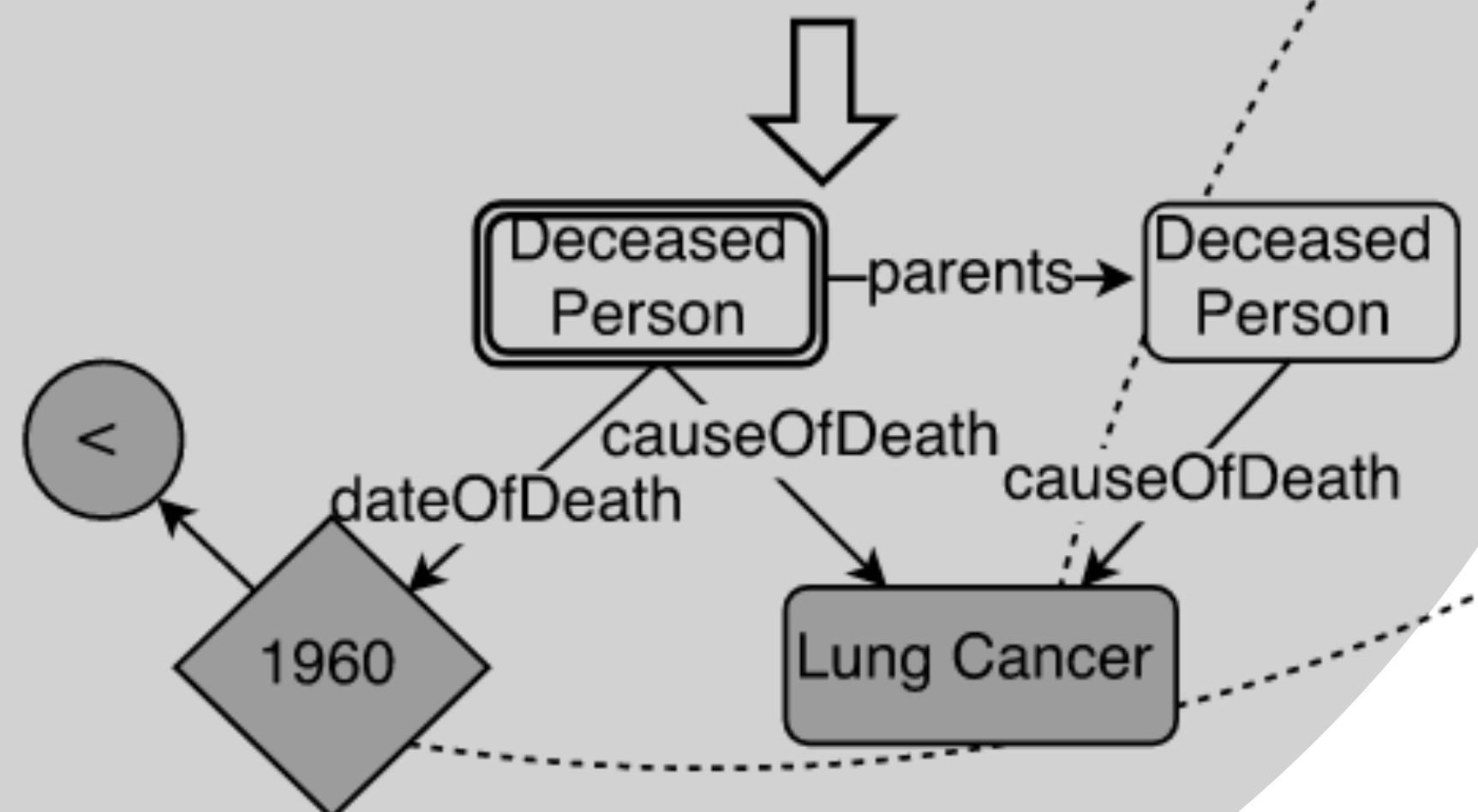
$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

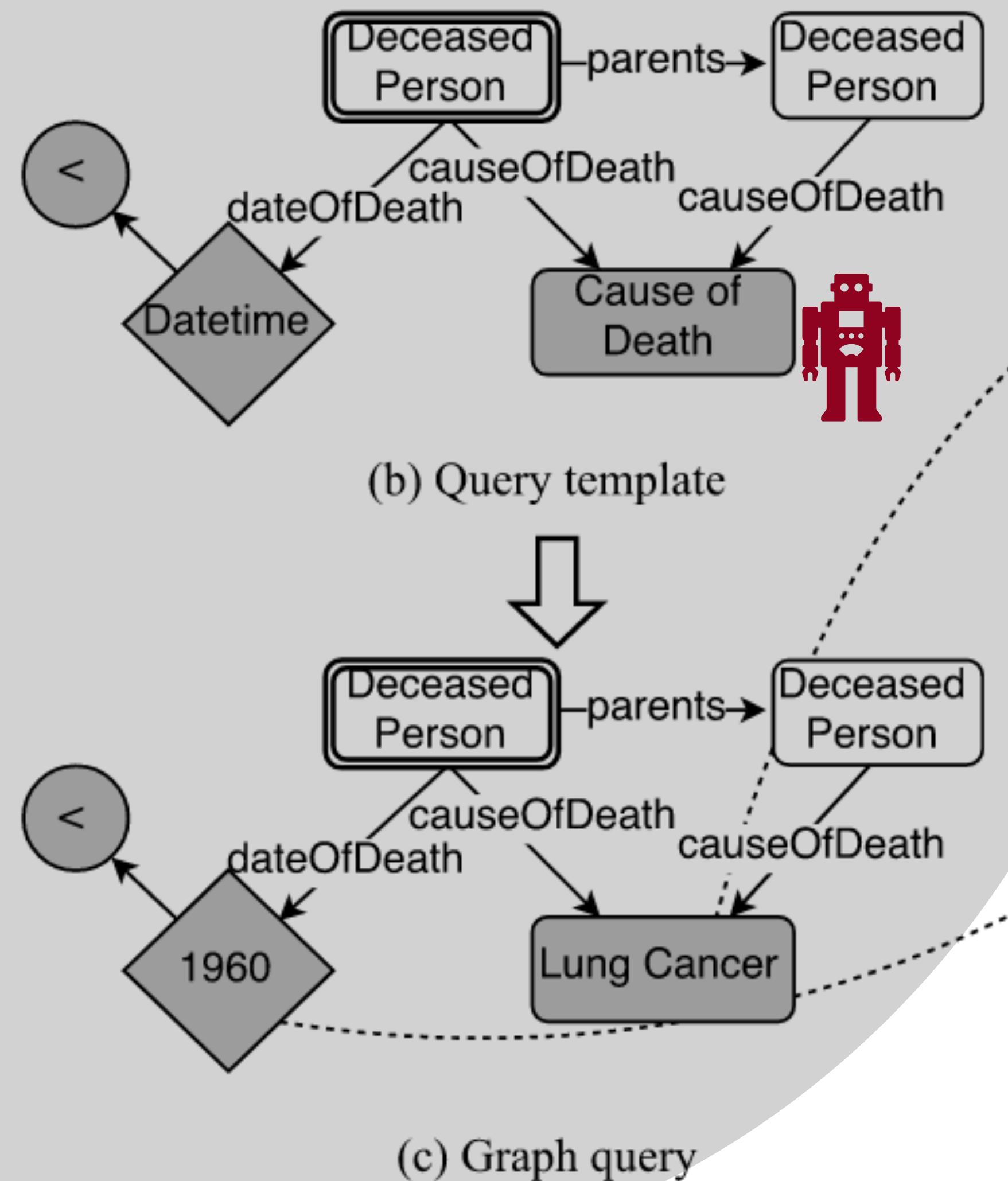
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

# Stage 1: Symbolic Graph Exploration



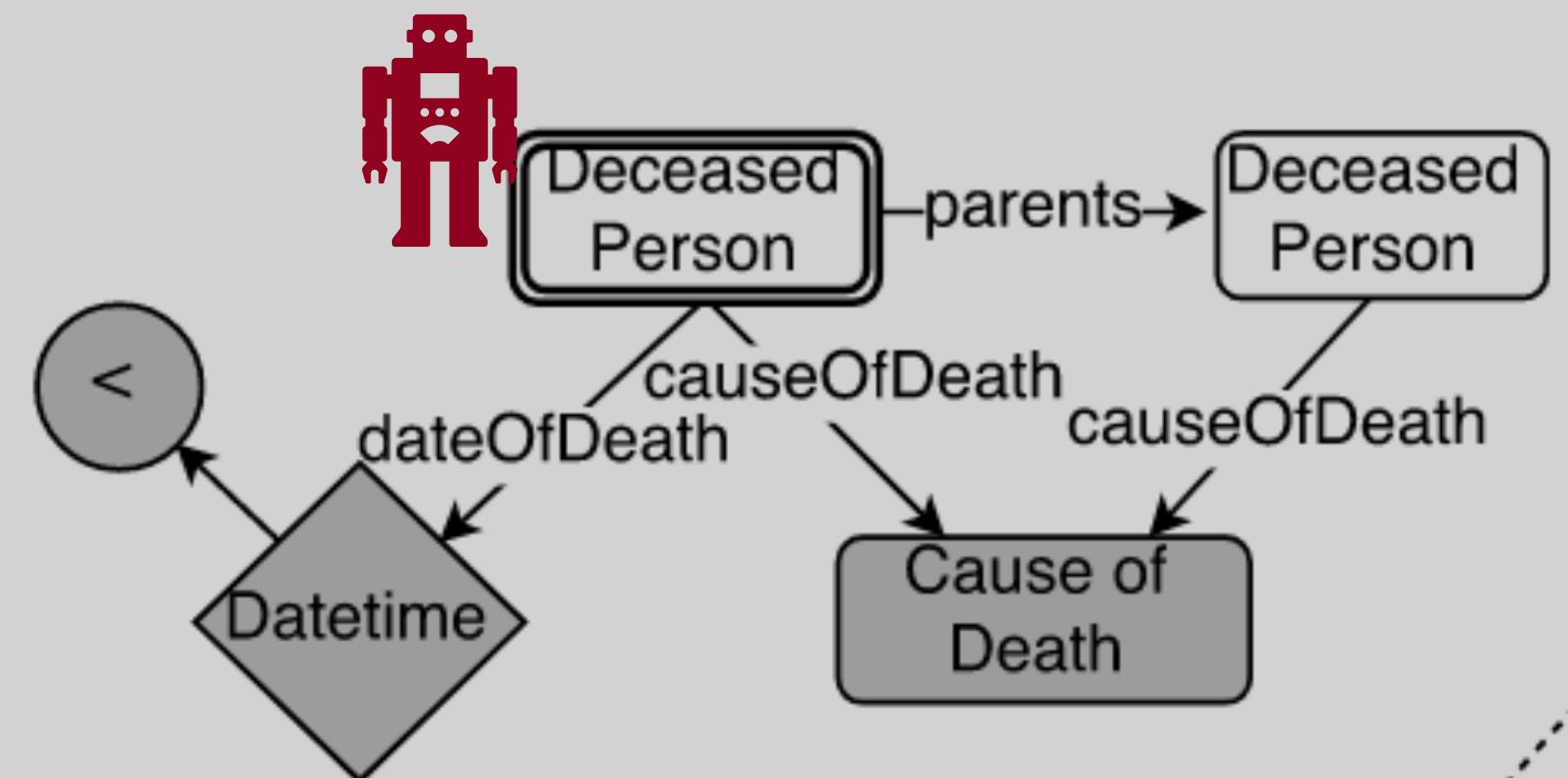
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

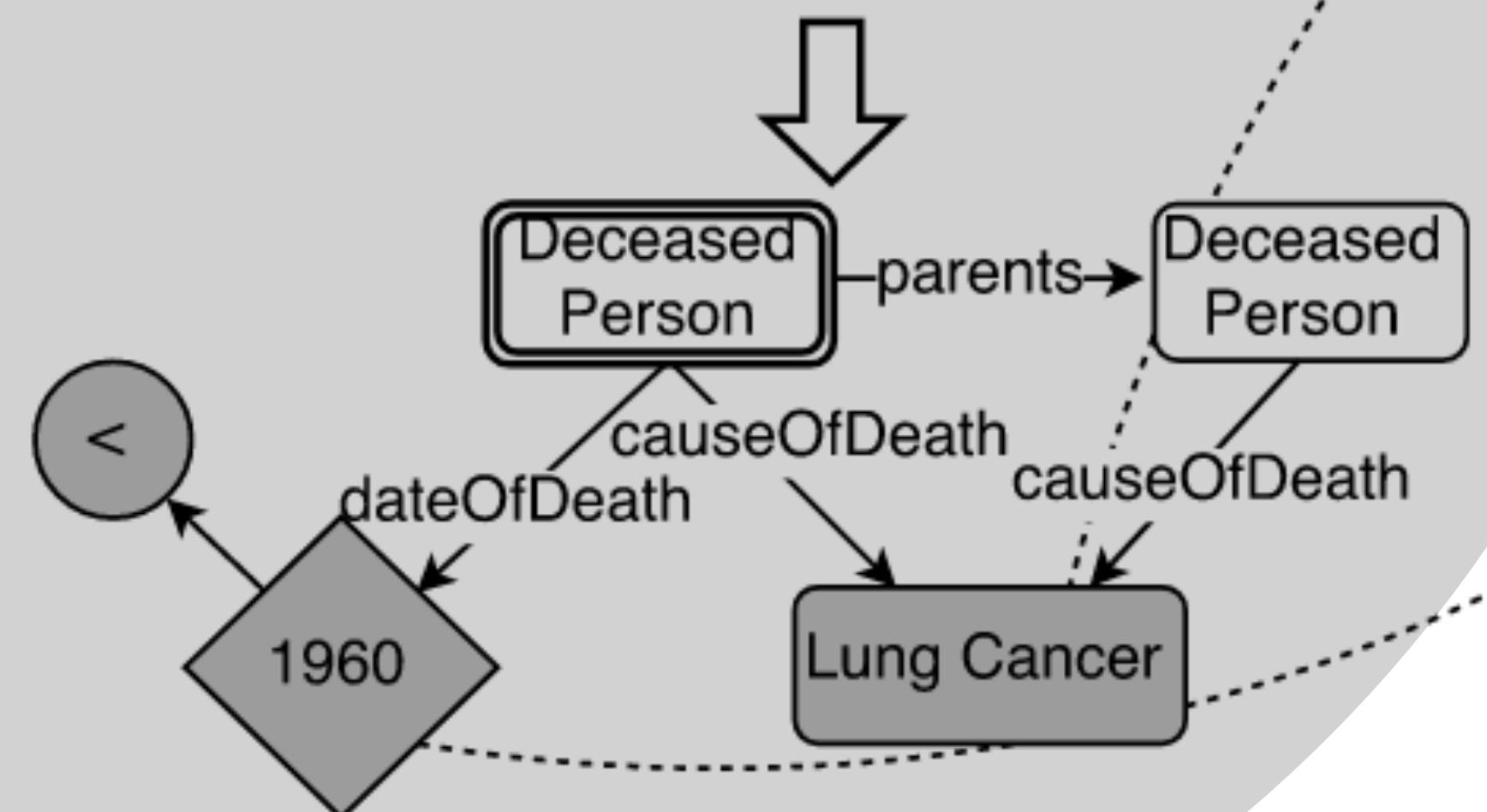
$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

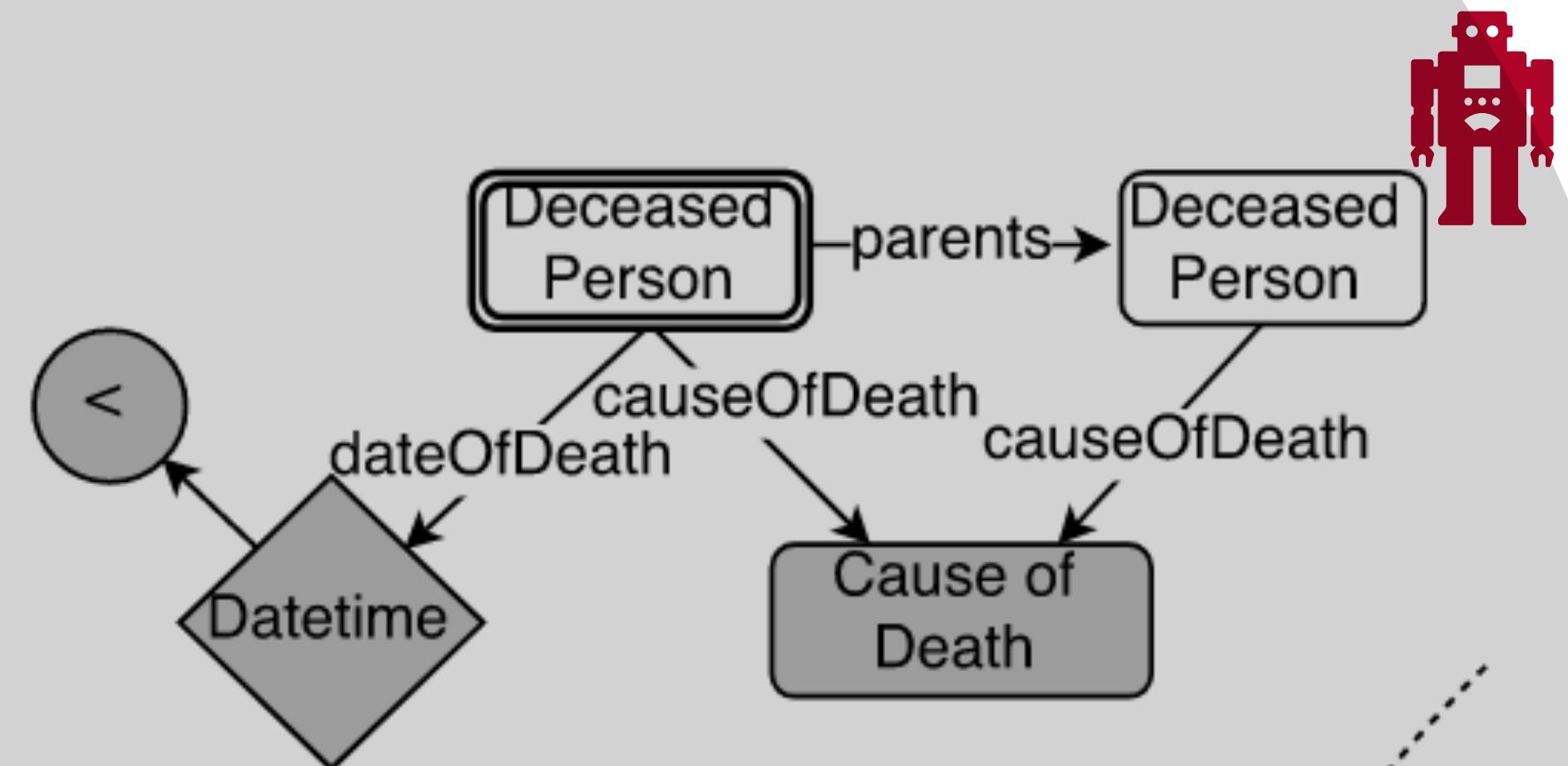
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

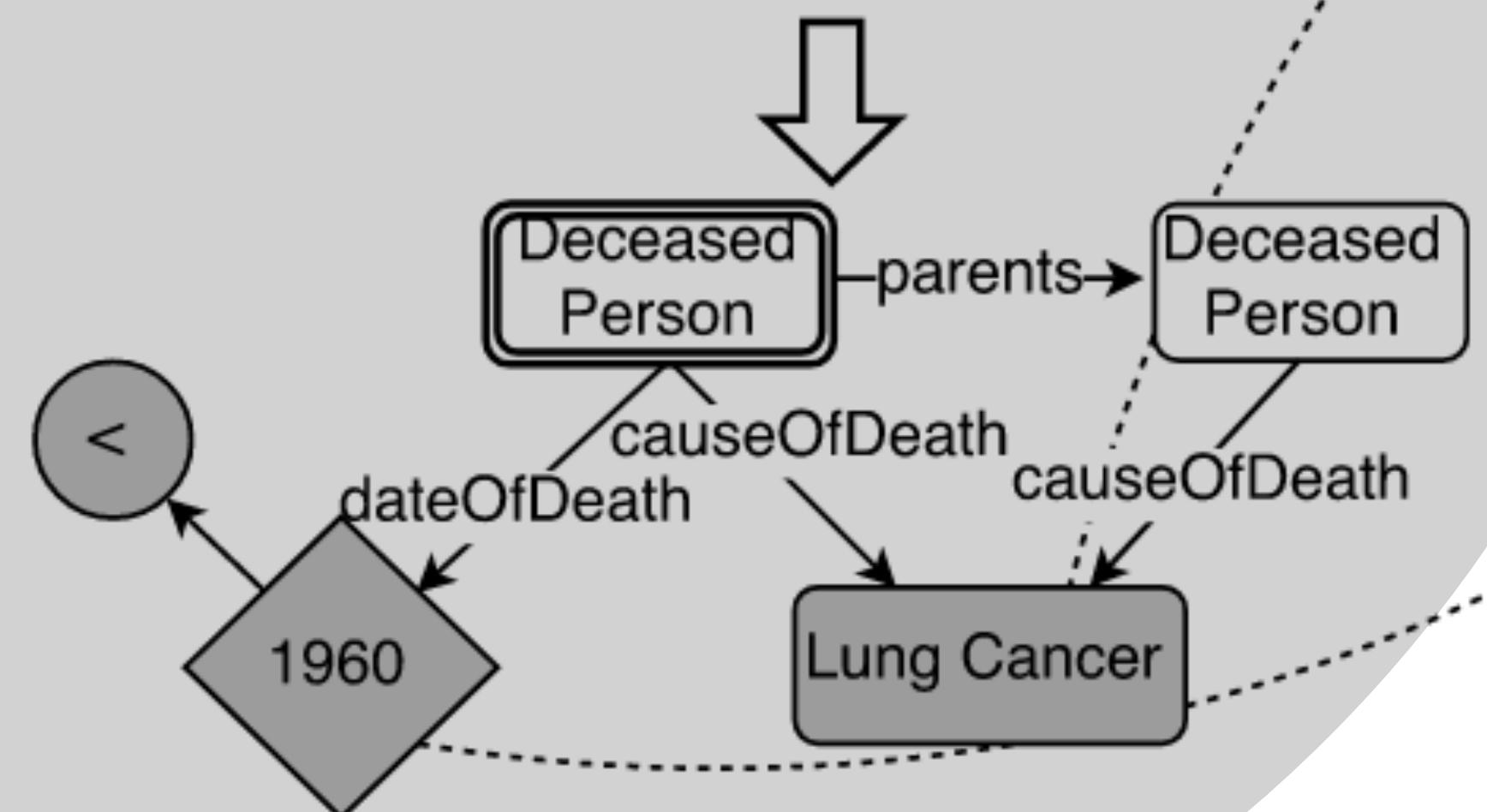
$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

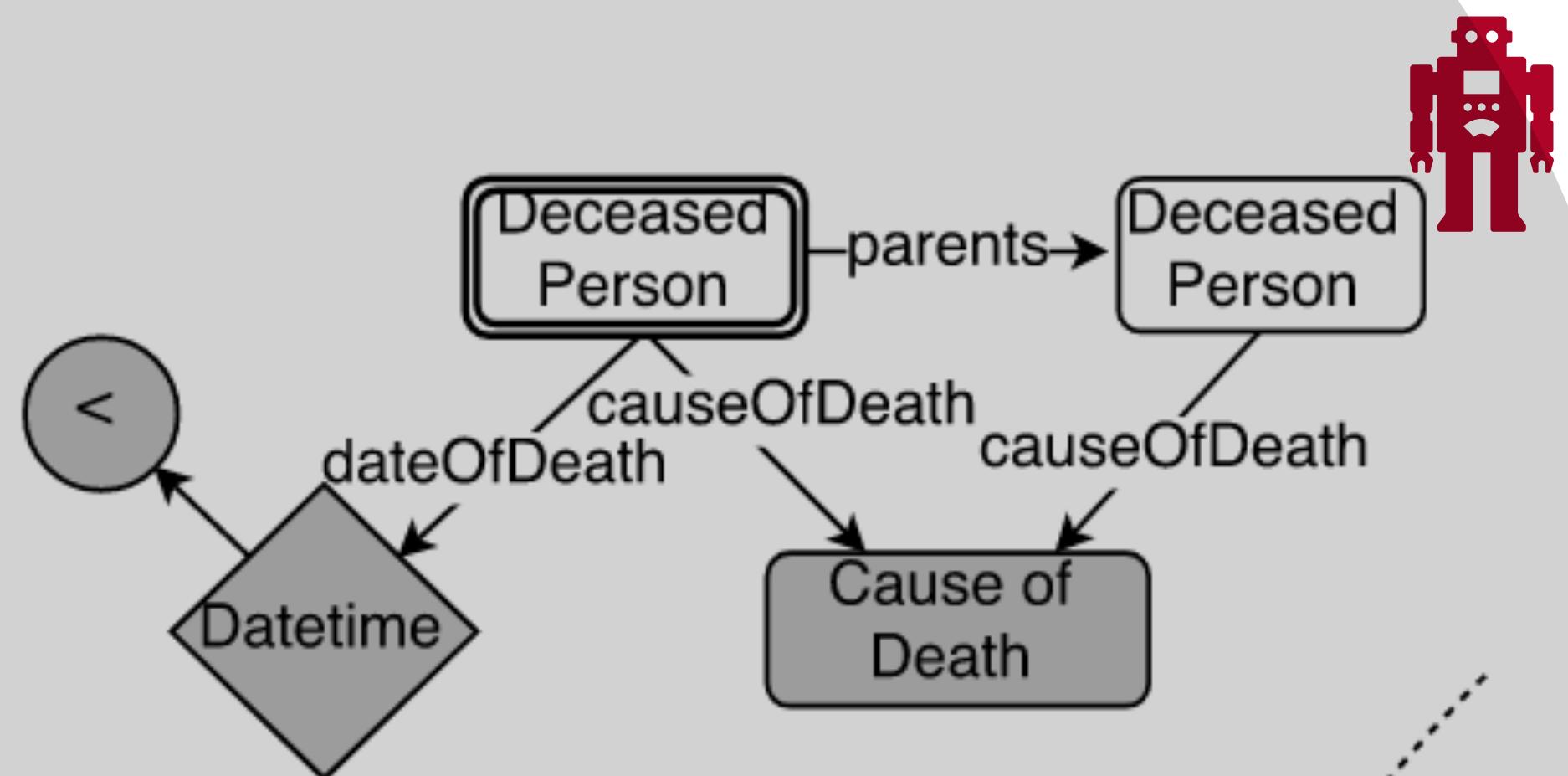
$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

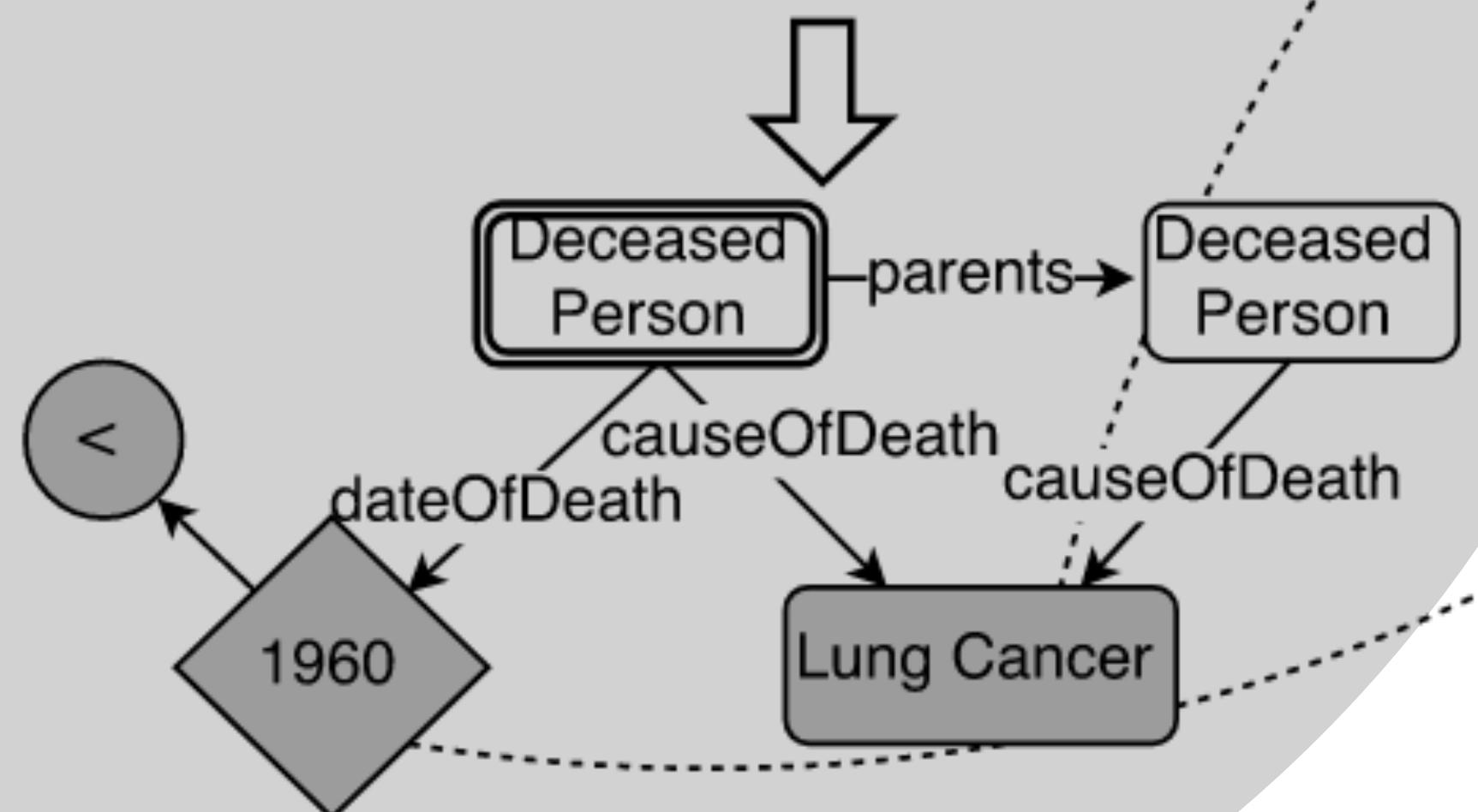
$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

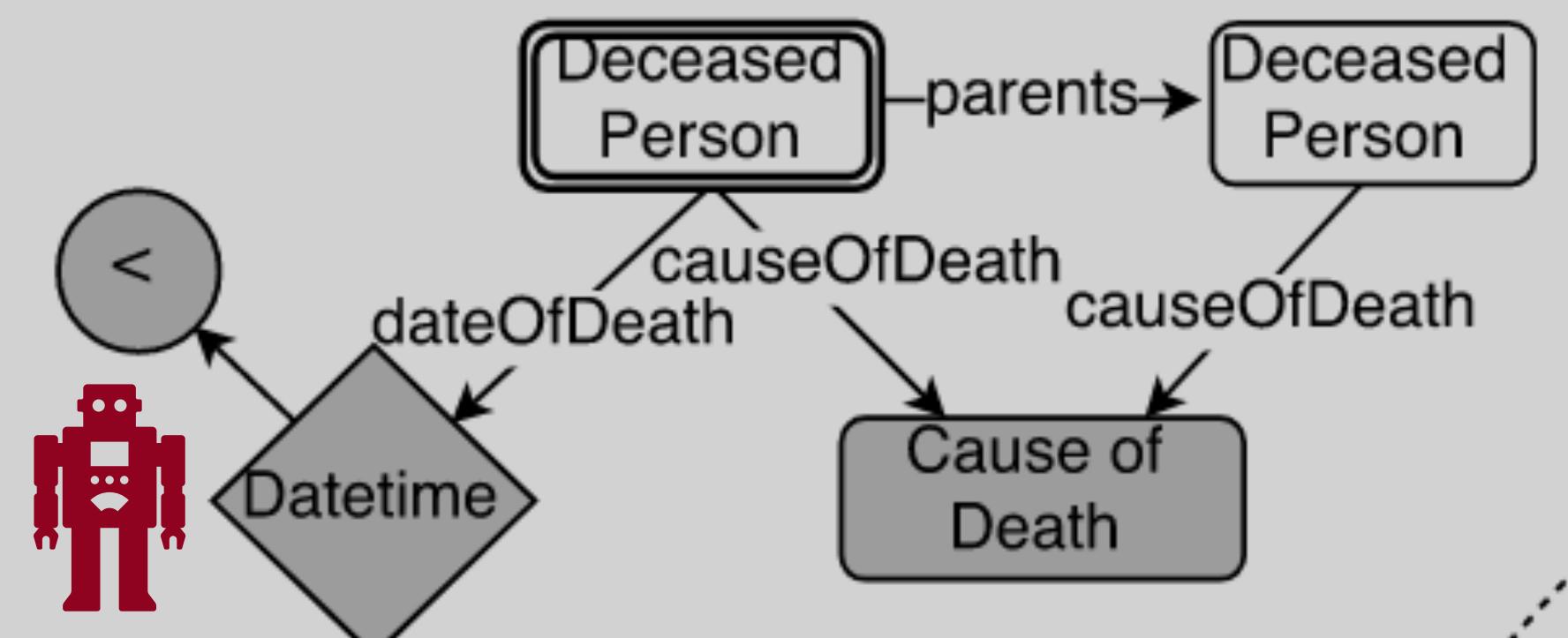
$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

$$p_1 := \text{extend}(p_0, s_0)$$

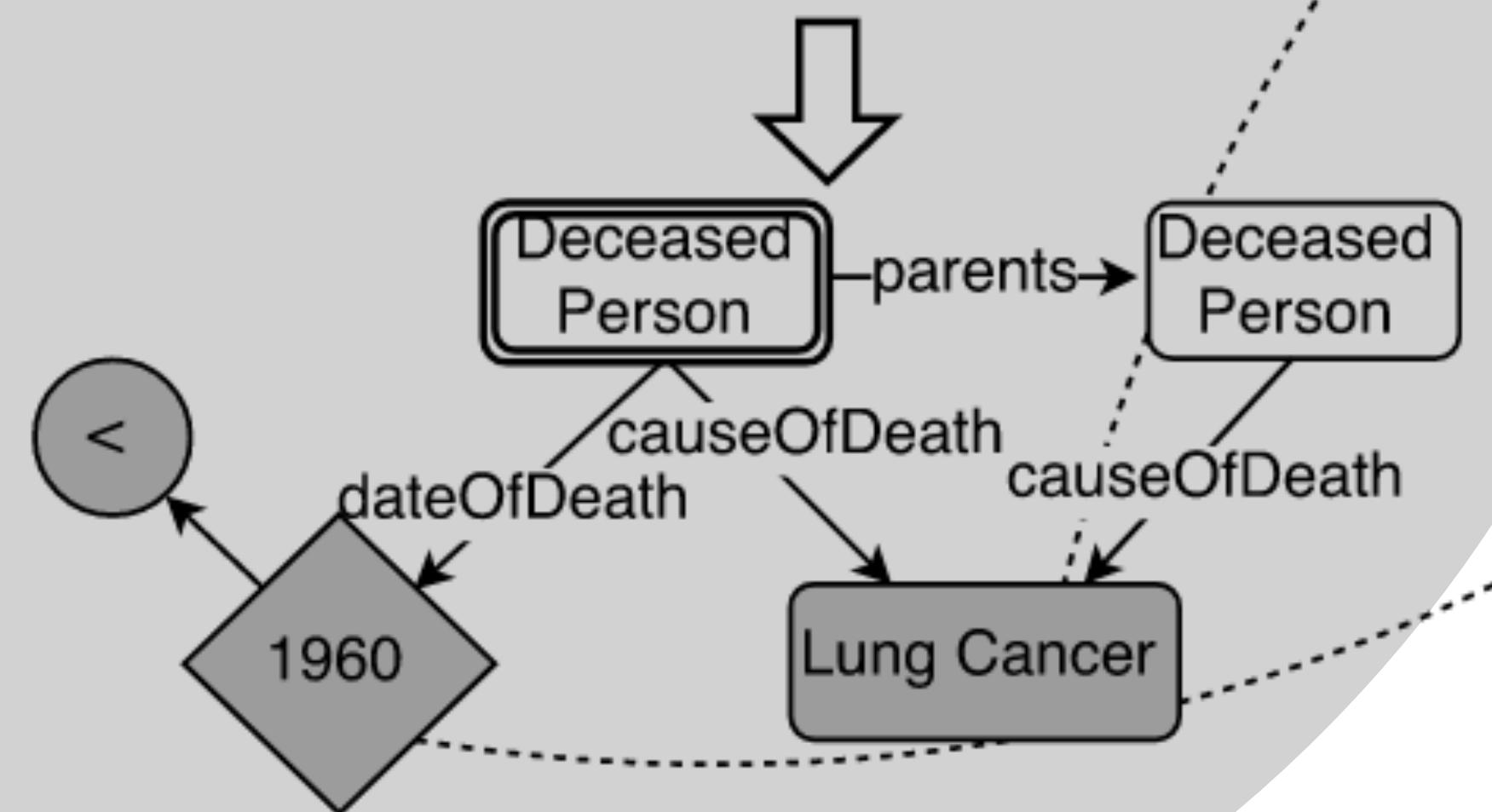
**Repeat till  $t$  for  $t$ -hop complexity**

$$f \sim \mathcal{F}$$

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

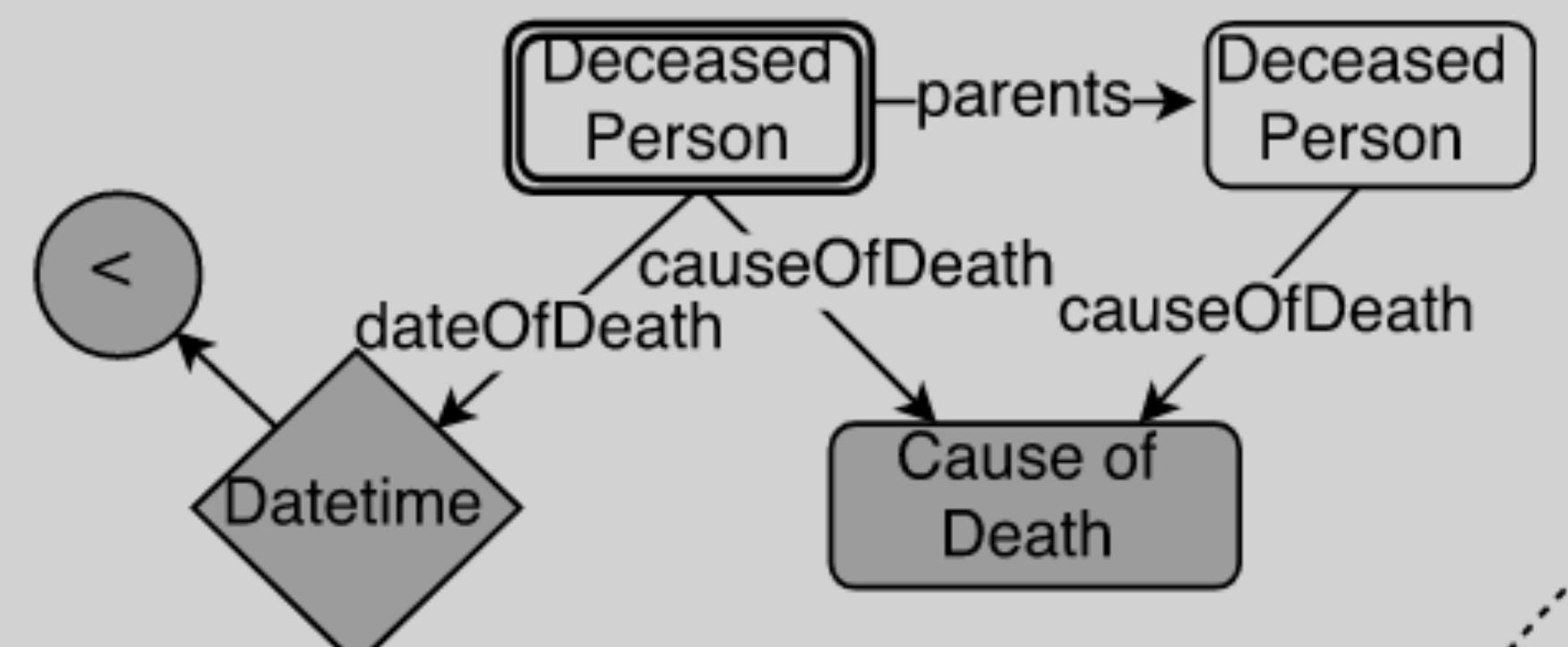
$$p_1 := \text{extend}(p_0, s_0)$$

**Repeat till  $t$  for  $t$ -hop complexity**

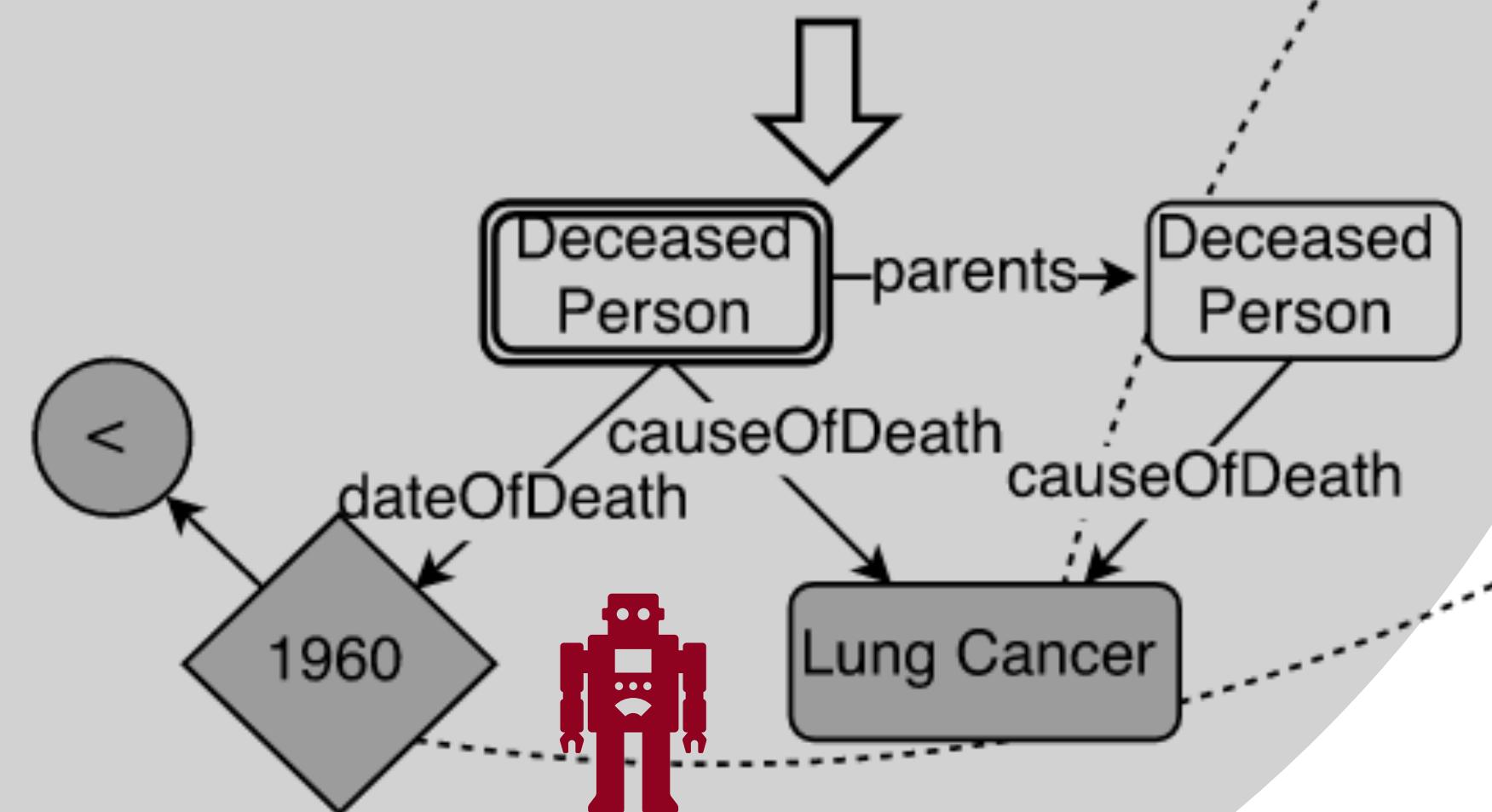
$$f \sim \mathcal{F}$$

$$p := \text{extend}(p_t, f)$$

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

$$p_1 := \text{extend}(p_0, s_0)$$

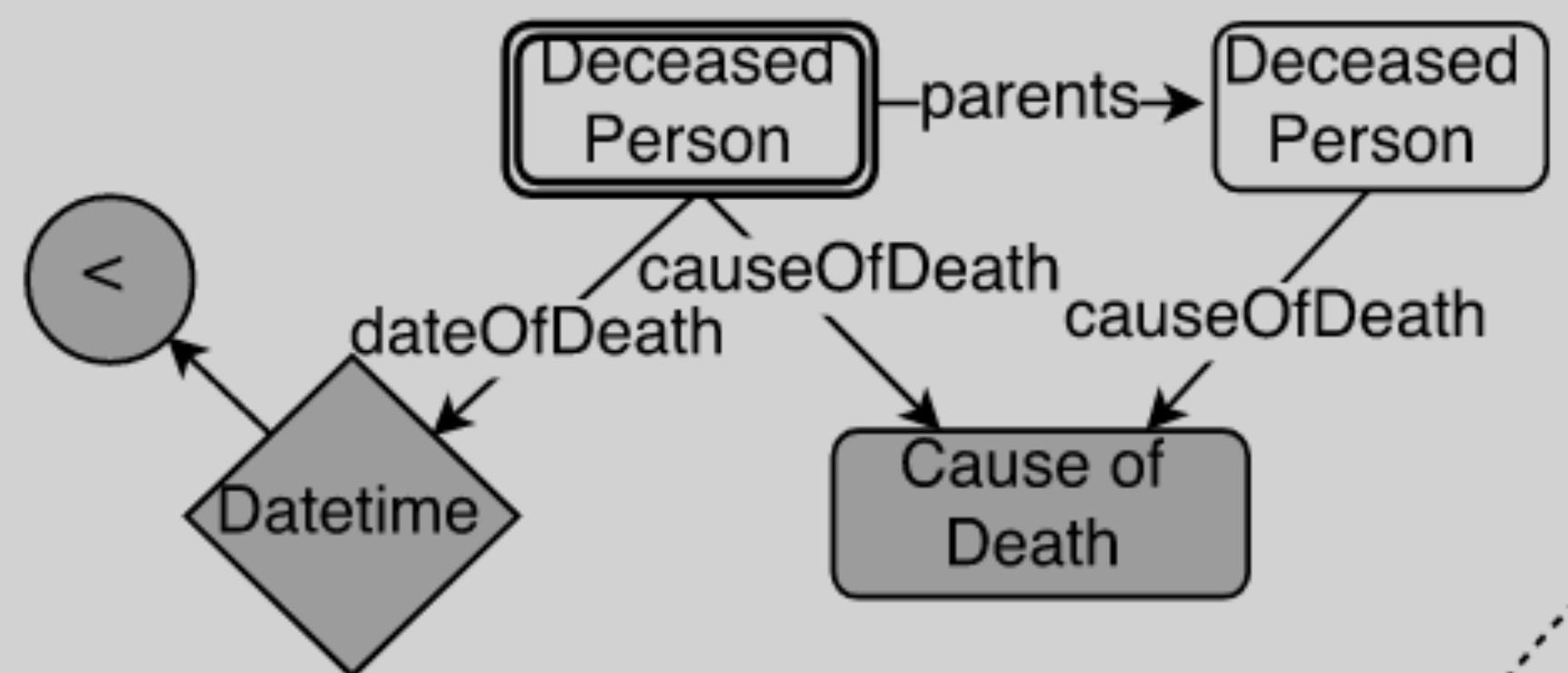
**Repeat till  $t$  for  $t$ -hop complexity**

$$f \sim \mathcal{F}$$

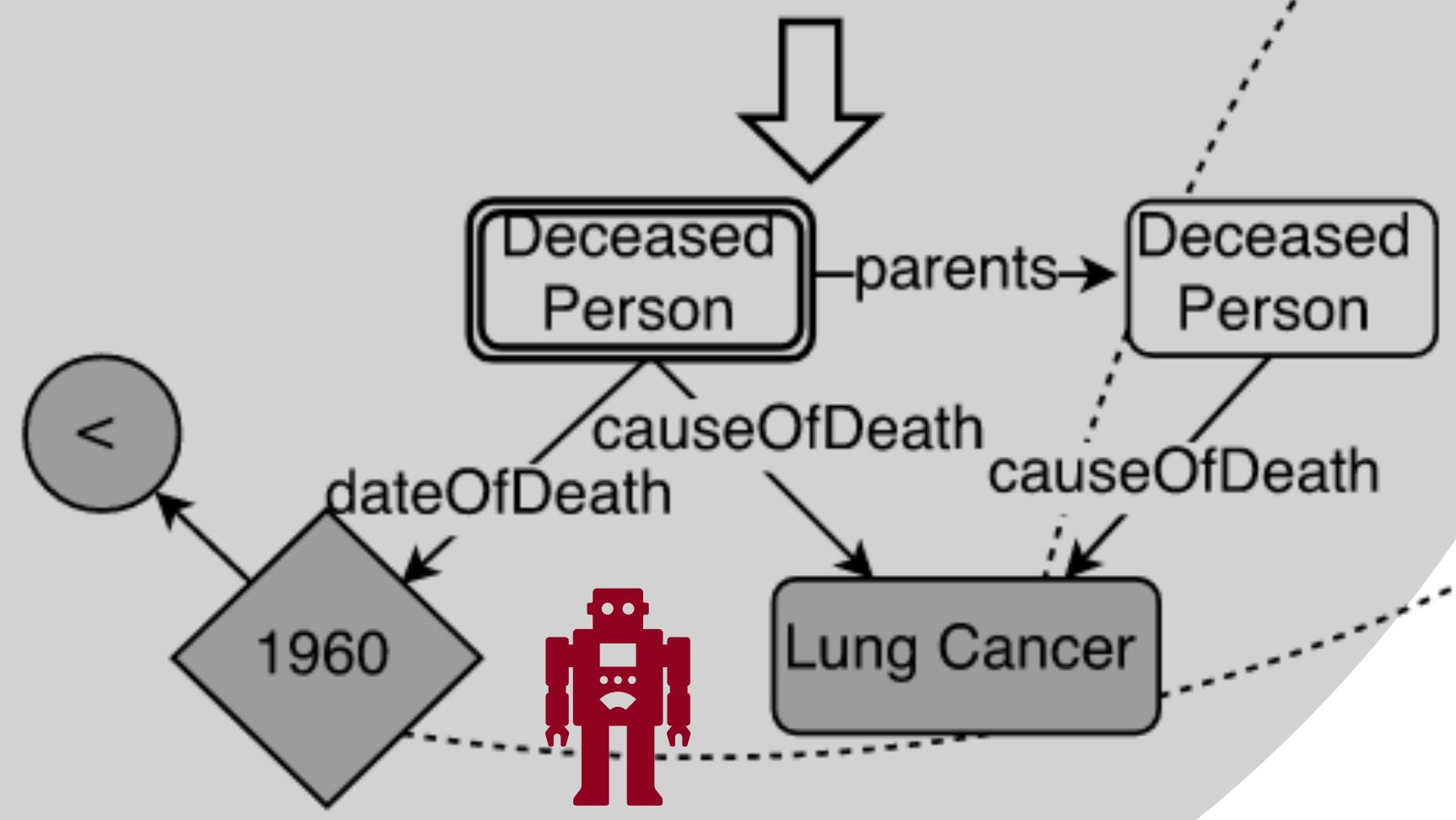
$$p := \text{extend}(p_t, f)$$

**Ground classes to entities**

# Stage 1: Symbolic Graph Exploration



(b) Query template



(c) Graph query

$$p_0 := c_o \sim \mathcal{C}$$

$$s_0 \sim \{s \mid s \in \mathcal{R} \cup \mathcal{C} : \text{reachable}(p_0, s)\}$$

$$p_1 := \text{extend}(p_0, s_0)$$

Repeat till  $t$  for  $t$ -hop complexity

$$f \sim \mathcal{F}$$

$$p := \text{extend}(p_t, f)$$

Ground classes to entities

Add  $p$  to  $\mathcal{X}^P$  **Program Exploration Set**

# Stage 1: Symbolic Graph Exploration

(AND medicine.manufactured\_drug\_form (JOIN medicine.manufactured\_drug\_form.marketing\_end\_date 2013-11-30^^http://www.w3.org/2001/XMLSchema#date) )

(AND biology.organism\_classification (JOIN biology.organism\_classification.fossil\_specimens m.0n8\_wf9) )

(AND measurement\_unit.measurement\_system (JOIN measurement\_unit.measurement\_system.pressure\_units m.0h5qxr7) )

(AND food.beer\_style\_category (JOIN food.beer\_style\_category.styles m.02hvlzv) )

(AND food.wine\_style (JOIN food.wine\_style.wines (JOIN wine.wine.wine\_producer m.03wz5rd) ) )

(COUNT (AND exhibitions.exhibition (JOIN (R exhibitions.exhibition\_producer.exhibitions\_produced) m.059wk) ))

(ARGMIN music.music\_video music.music\_video.initial\_release\_date)

# Stage 1: Symbolic Graph Exploration

```
(AND medicine.manufactured_drug_form (JOIN medicine.manufactured_drug_form.marketing_end_date 2013-11-30^^http://www.w3.org/2001/XMLSchema#date) )
```

```
(AND biology.organism_classification (JOIN biology.organism_classification.fossil_specimens m.0n8_wf9) )
```

```
(AND measurement_unit.measurement_system (JOIN measurement_unit.measurement_system.pressure_units m.0h5qxr7) )
```

```
(AND food.beer_style_category (JOIN food.beer_style_category.styles m.02hvlzv) )
```

```
(AND food.wine_style (JOIN food.wine_style.wines (JOIN wine.wine.wine_producer m.03wz5rd) ) )
```

```
(COUNT (AND exhibitions.exhibition (JOIN (R exhibitions.exhibition_producer.exhibitions_produced) m.059wk) ))
```

```
(ARGMIN music.music_video music.music_video.initial_release_date)
```

# Stage 1: Symbolic Graph Exploration

```
(AND medicine.manufactured_drug_form (JOIN medicine.manufactured_drug_form.marketing_end_date 2013-11-30^^http://www.w3.org/2001/XMLSchema#date) )
```

```
(AND biology.organism_classification (JOIN biology.organism_classification.fossil_specimens m.0n8_wf9) )
```

```
(AND measurement_unit.measurement_system (JOIN measurement_unit.measurement_system.pressure_units m.0h5qxr7) )
```

```
(AND food.beer_style_category (JOIN food.beer_style_category.styles m.02hvlzv) )
```

```
(AND food.wine_style (JOIN food.wine_style.wines (JOIN wine.wine.wine_producer m.03wz5rd) ) )
```

```
(COUNT (AND exhibitions.exhibition (JOIN (R exhibitions.exhibition_producer.exhibitions_produced) m.059wk) ) )
```

```
(ARGMIN music.music_video music.music_video.initial_release_date)
```

# Stage 1: Symbolic Graph Exploration

```
(AND medicine.manufactured_drug_form (JOIN medicine.manufactured_drug_form.marketing_end_date 2013-11-30^^http://www.w3.org/2001/XMLSchema#date) )
```

```
(AND biology.organism_classification (JOIN biology.organism_classification.fossil_specimens m.0n8_wf9) )
```

**Fast! 10,000 programs in ~1.5hrs on Freebase**

```
(AND food.wine_style (JOIN food.wine_style.wines (JOIN wine.wine.wine_producer m.03wz5rd) )
```

```
(COUNT (AND exhibitions.exhibition (JOIN (R exhibitions.exhibition_producer.exhibitions_produced) m.059wk) ))
```

```
(ARGMIN music.music_video music.music_video.initial_release_date)
```

# Stage 2: Question Generation

**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$

# Stage 2: Question Generation

**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$

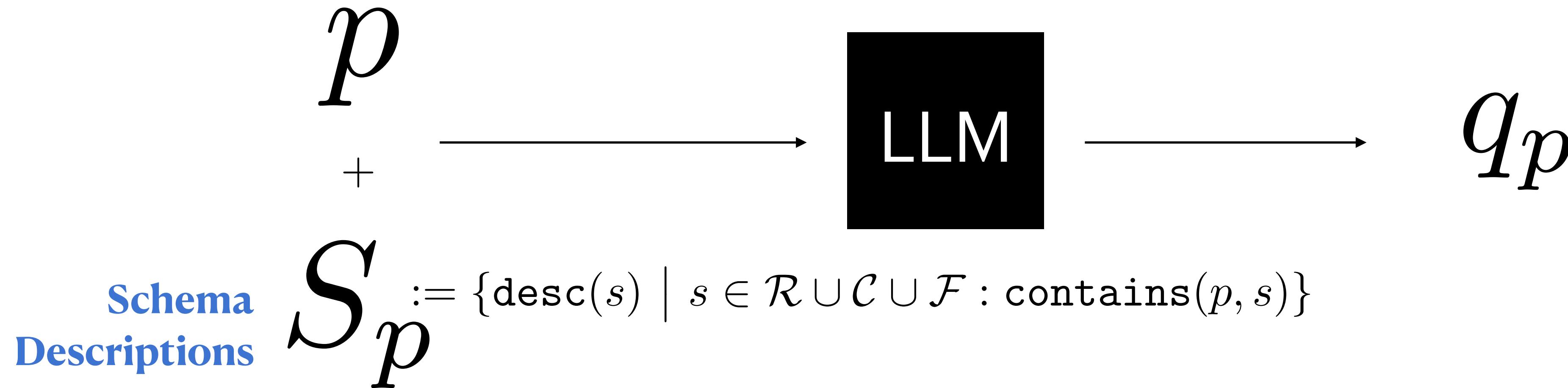
# Stage 2: Question Generation

**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$



# Stage 2: Question Generation

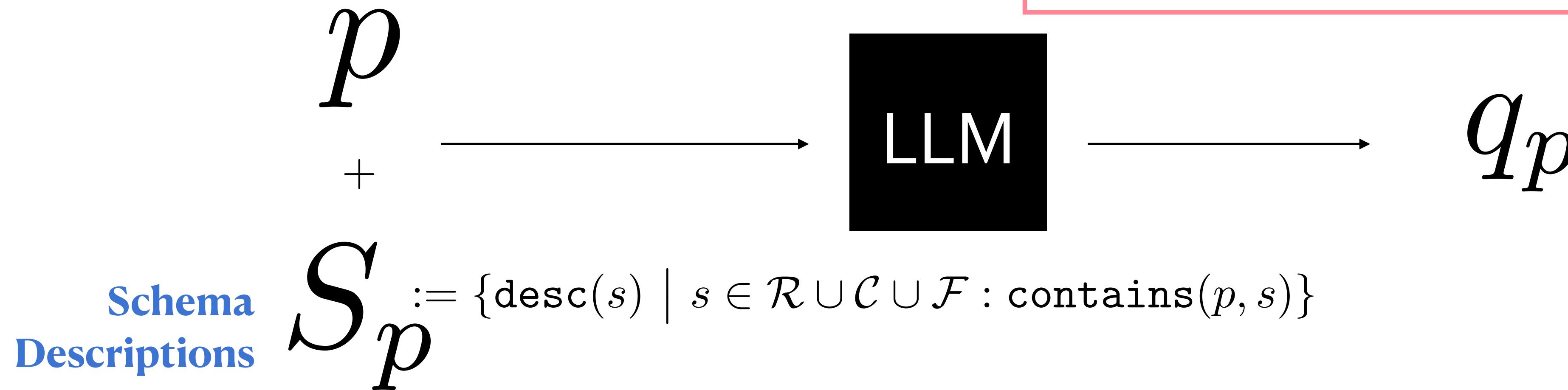
**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$



# Stage 2: Question Generation

**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$

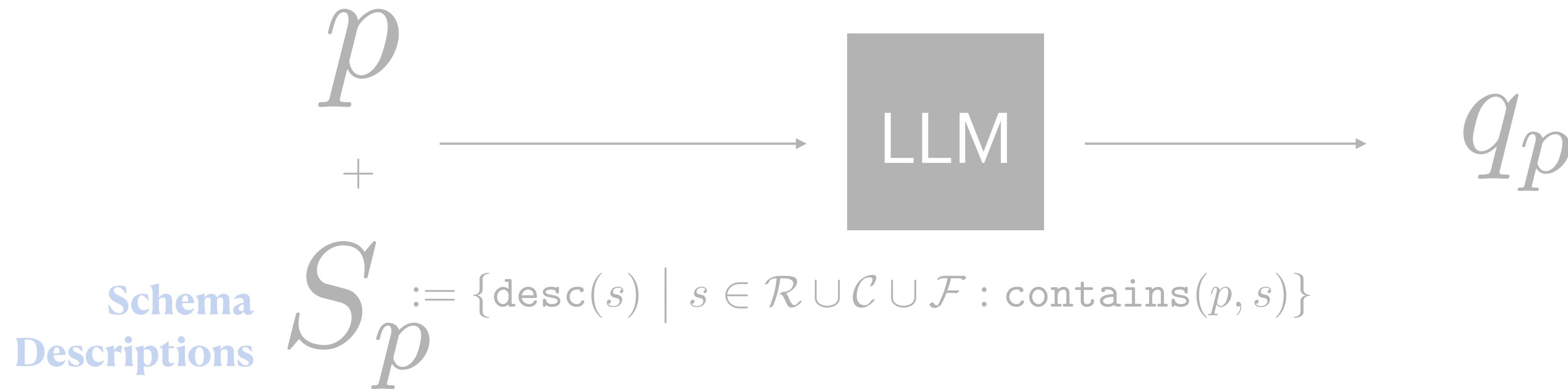
1. Most of our experiments are with small-mid LLMs (7B)
2. We use open-source models.



# Stage 2: Question Generation

**Goal:**  $\mathcal{X} := \{(q_p, p \mid p \in \mathcal{X}^P)\}$

*where*



# Stage 2: Question Generation

Goal.  $\chi ::= \langle a_n \mid n \in \chi^P \rangle \cup$

However, zero-shot LLM generation is challenging:

- #1 Erroneous generations for complex, multi-hop programs
- #2 Incorrect top-1 predictions from the model

Schema  
Descriptions

$$S_p := \{\text{desc}(s) \mid s \in \mathcal{R} \cup \mathcal{C} \cup \mathcal{F} : \text{contains}(p, s)\}$$

# Stage 2: Question Generation

## Solution for #1

### Least-to-Most Prompting

**Query:** (JOIN (R movie.written\_by) (JOIN movie.starred\_actors (JOIN (R movie.starred\_actors) "Titanic")))

**L2M prompting:**

# Step 1:  
(JOIN (R movie.starred\_actors) m.6594)

**Prediction:** Who starred in Titanic?

# Step 2:  
(JOIN (R movie.starred\_actors) m.6594)  
Who starred in Titanic?

(JOIN movie.starred\_actors (JOIN (R movie.starred\_actors) "Titanic"))

**Prediction:** What movies have the actors who starred in Titanic starred in?

# Step 3:  
(JOIN (R movie.starred\_actors) m.6594)  
Who starred in Titanic?

(JOIN movie.starred\_actors (JOIN (R movie.starred\_actors) "Titanic"))

What movies have the actors who starred in Titanic starred in?

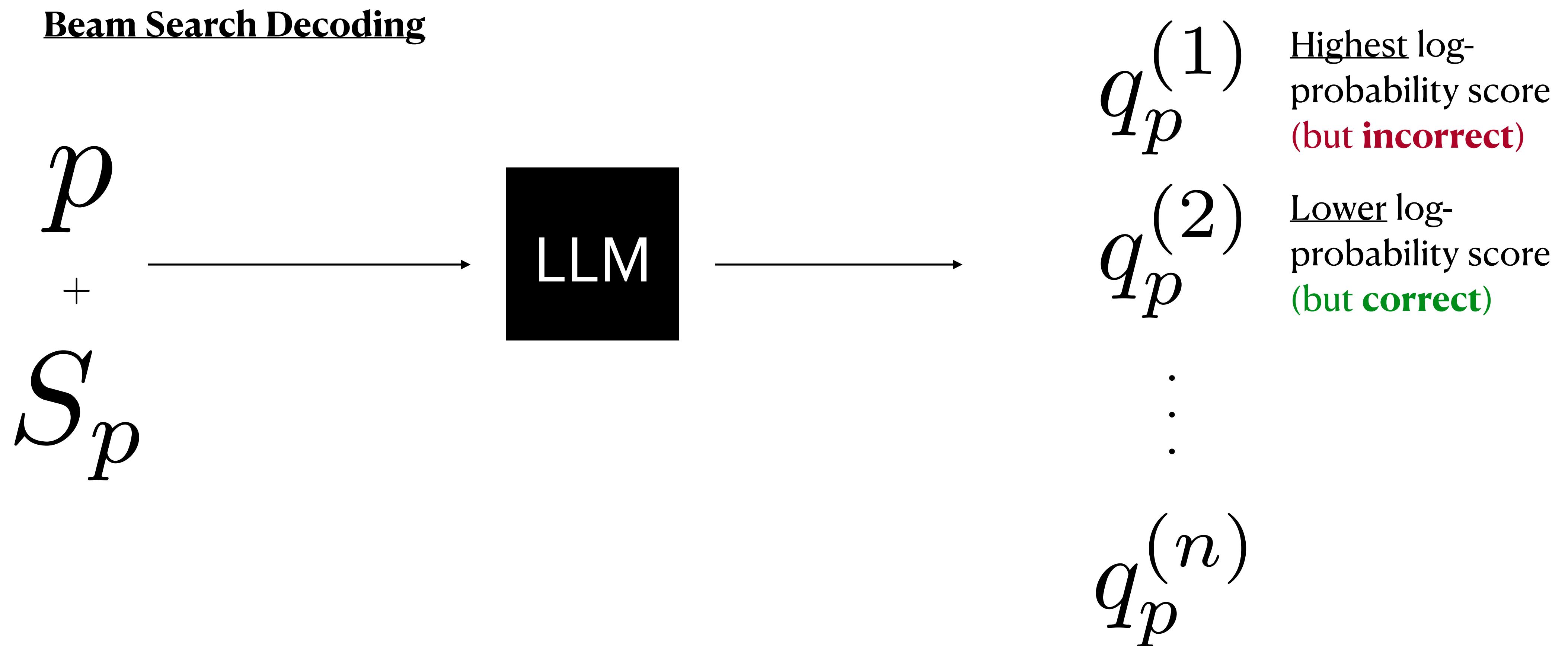
(JOIN (R movie.written\_by) (JOIN movie.starred\_actors (JOIN (R movie.starred\_actors) "Titanic")))

**Prediction:** Who wrote the movies that have actors who<sup>49</sup> starred in the Titanic?

# Stage 2: Question Generation

## Problem #2

**Beam Search Decoding**



# Stage 2: Question Generation



Solution for #2

**Inverse-Consistency Re-ranking**

Given a generative task

$$T := y \mid I, D, x$$

For our setting,  $D = \emptyset$

Standard top-1 prediction  
from a decoding algorithm

$$y_{\text{pred}} := \arg \max_{y \in \mathbf{y}_{\text{cands}}} \log \Pr(y \mid I, D, x) / |y|$$

Score using the *inverse* task

$$T^{-1} := x \mid I^{-1}, D^{-1}, y$$

$$y_{\text{pred}} := \arg \max_{y \in \mathbf{y}_{\text{cands}}} \log \Pr(x \mid I^{-1}, D^{-1}, y) / |x|$$

# Stage 2: Question Generation



Solution for #2

## Inverse-Consistency Re-ranking

```
"query": "(AND religion.founding_figure (JOIN religion.founding_figure.religion Founded  
(JOIN religion.religion.founding_figures \"st. peter\")))",  
"reference": "who were the founding figures of the religion that st. peter founded"  
"standard_decoding": [  
    "who is paul the apostle?",  
    "who founded christianity?",  
    "who was the founder of christianity?",  
    "what is the name of the person who founded christianity?",  
    "who are the founding figures of the religion founded by st. peter?",  
    "who is the founding figure of the religion founded by st. peter?",  
]  
"inverse_consistency": [  
    "who is the founding figure of the religion founded by st. peter?",  
    "who are the founding figures of the religion founded by st. peter?",  
    "who was the founder of christianity?",  
    "who founded christianity?",  
    "who is paul the apostle?",  
    "what is the name of the person who founded christianity?",  
]
```

# Stage 2: Question Generation



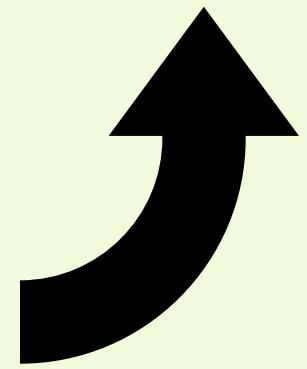
Solution for #2

## Inverse-Consistency Re-ranking

```
"query": "(AND religion.founding_figure (JOIN religion.founding_figure.  
(JOIN religion.religion.founding_figures \"st. peter\"))))",
```

Necessary for fast inference  
since question generation is  
the bottleneck in exploration

Reliably improves top-1 prediction!  
Particularly for models with **lower-parameter** counts.



```
"inverse_consistency": [  
    "who is the founding figure of the religion founded by st. peter?",  
    "who are the founding figures of the religion founded by st. peter?",  
    "who was the founder of christianity?",  
    "who founded christianity?",  
    "who is paul the apostle?",  
    "what is the name of the person who founded christianity?",  
]
```

# We have Q&A pairs, lets train a model!

# We have Q&A pairs, lets train a model!

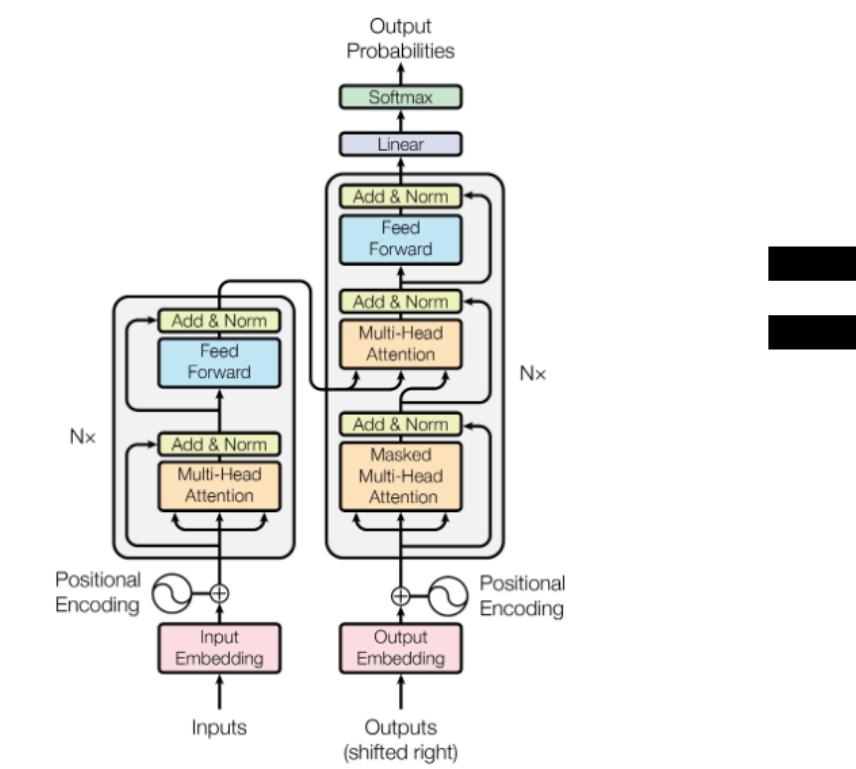
- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

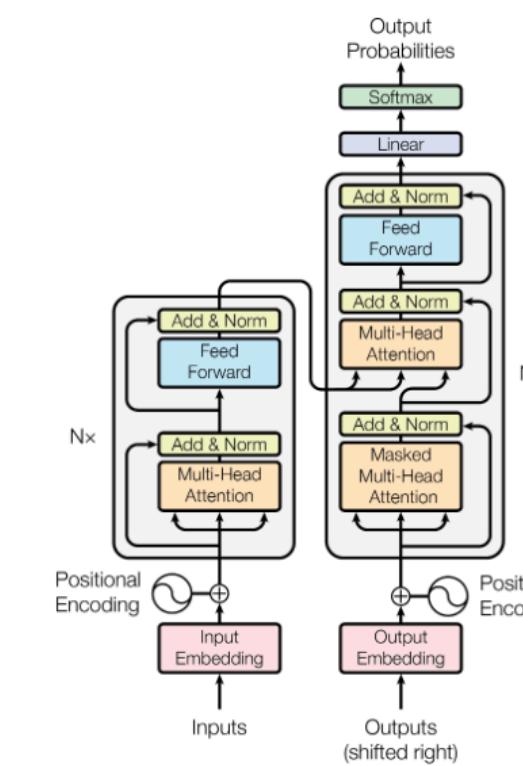
A parametric model  
specific to the KG!

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

A parametric model  
specific to the KG!

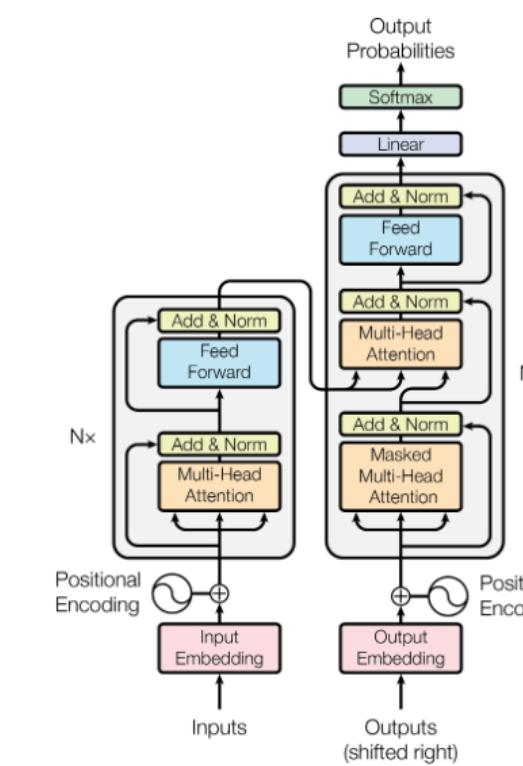
Hence will not work for other KGs 😢

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

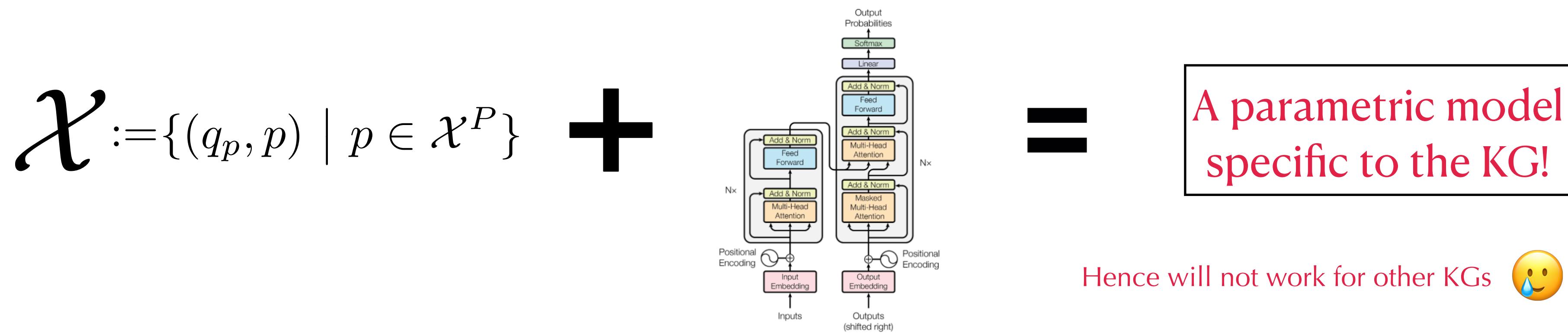
A parametric model  
specific to the KG!

Hence will not work for other KGs 😢

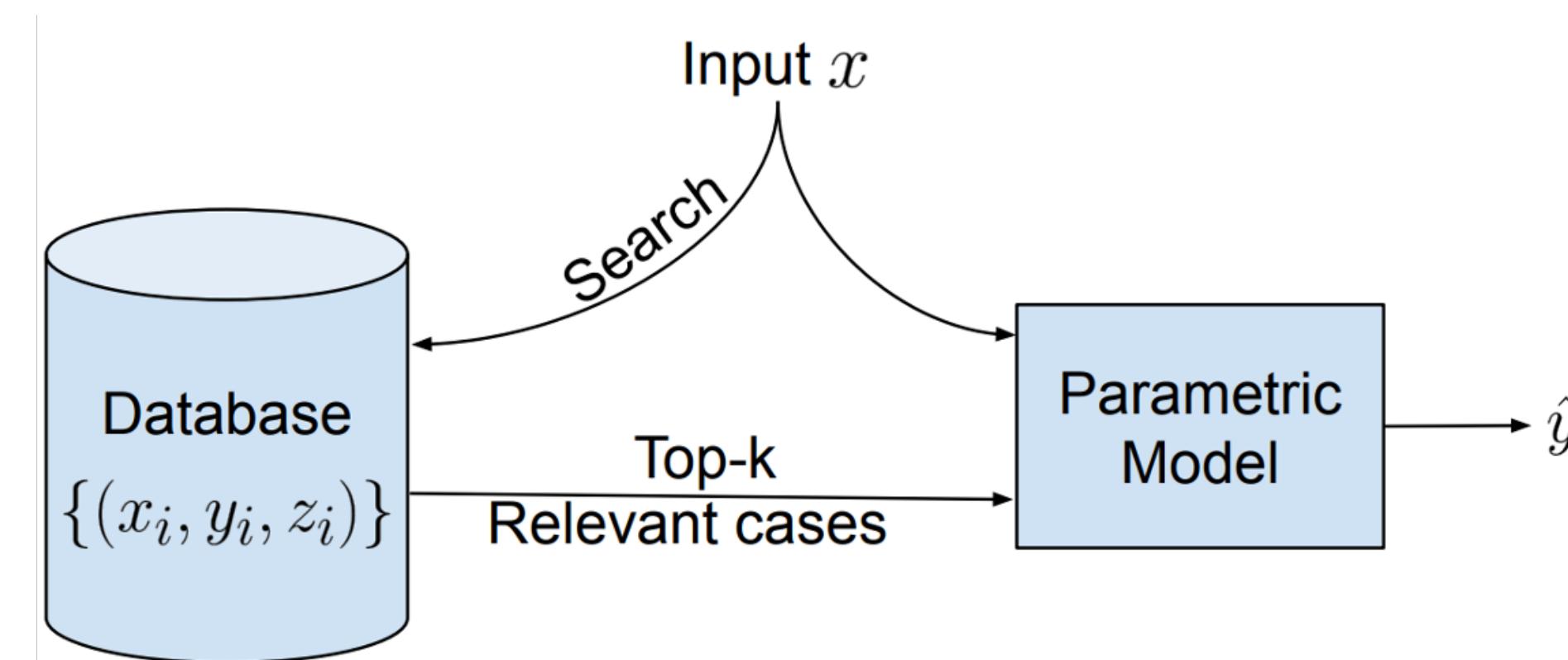
- Semiparametric approach with the generated (Q,P) pairs in nonparametric memory.

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?



- Semiparametric approach with the generated (Q,P) pairs in nonparametric memory.

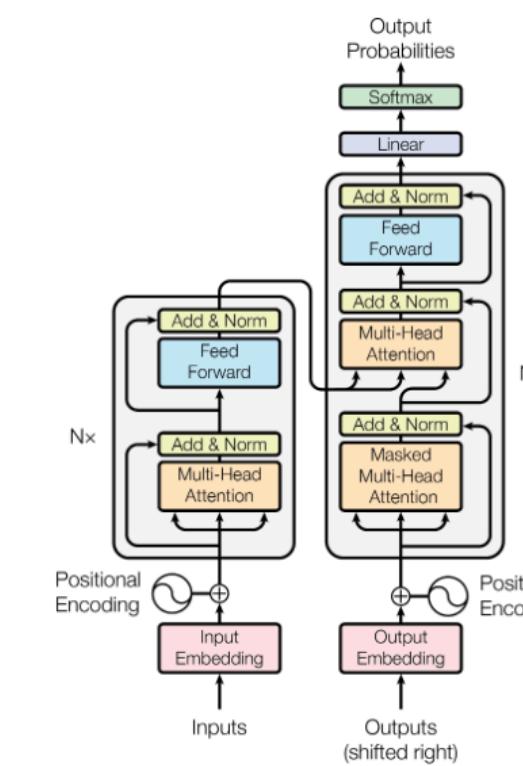


# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

A parametric model  
specific to the KG!

Hence will not work for other KGs 😢

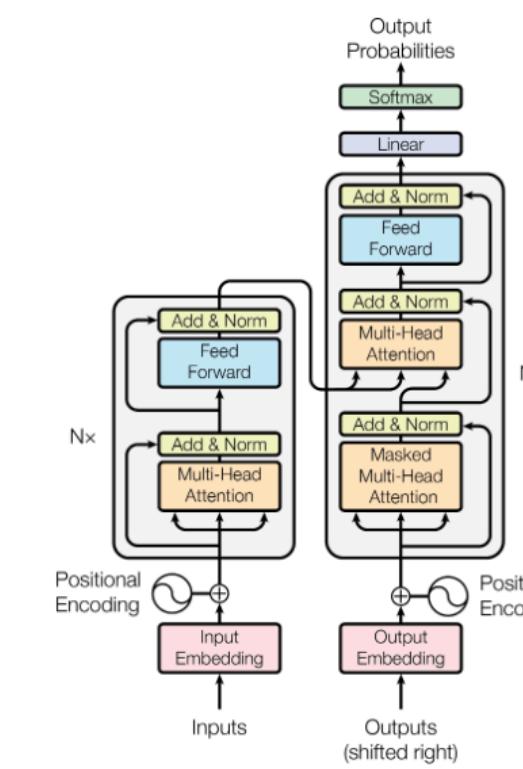
- Semiparametric approach with the generated (Q,P) pairs in nonparametric memory.

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

A parametric model  
specific to the KG!

Hence will not work for other KGs 😢

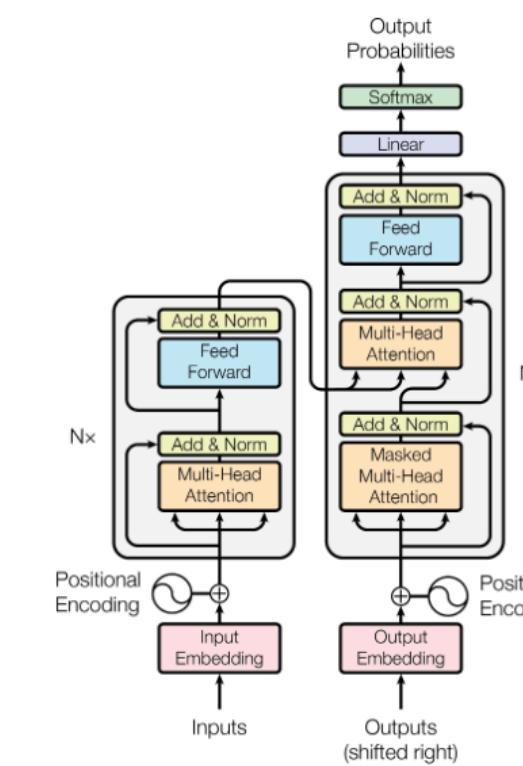
- Semiparametric approach with the generated (Q,P) pairs in nonparametric memory.
- The parametric component is an LLM that uses the demonstration retrieved from the memory

# We have Q&A pairs, lets train a model!

- We have query and program pairs that we generated from the graph. So we can now train supervised model on this graph, right?

$$\mathcal{X} := \{(q_p, p) \mid p \in \mathcal{X}^P\}$$

+



=

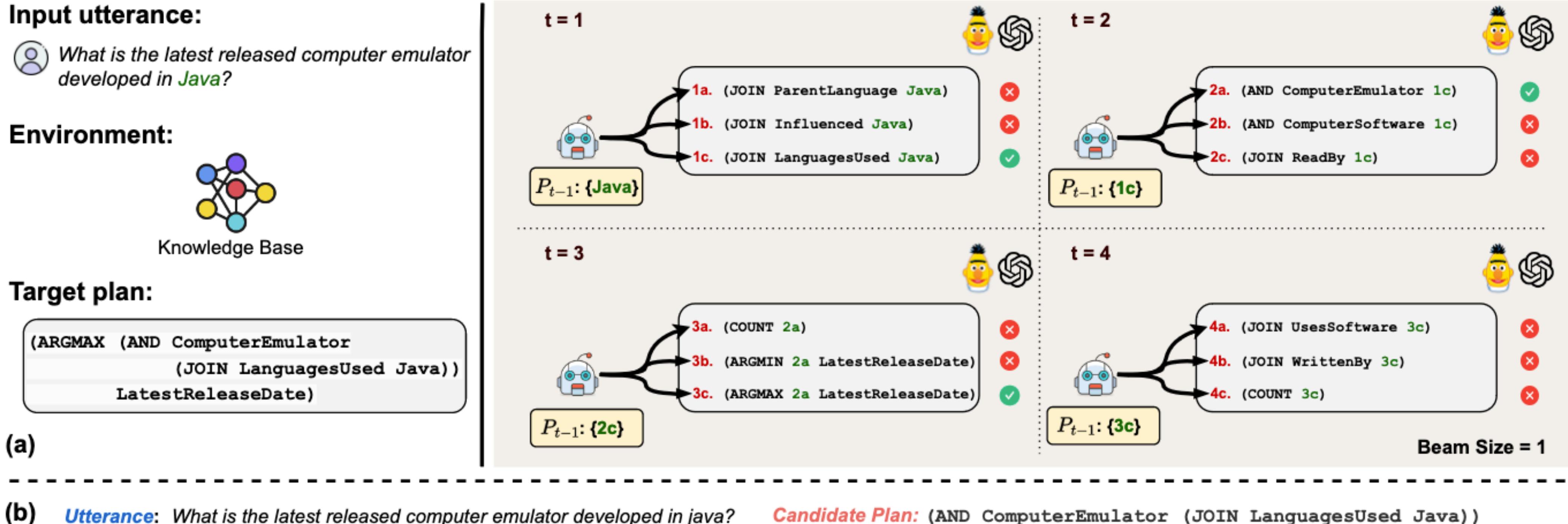
A parametric model  
specific to the KG!

Hence will not work for other KGs 😢

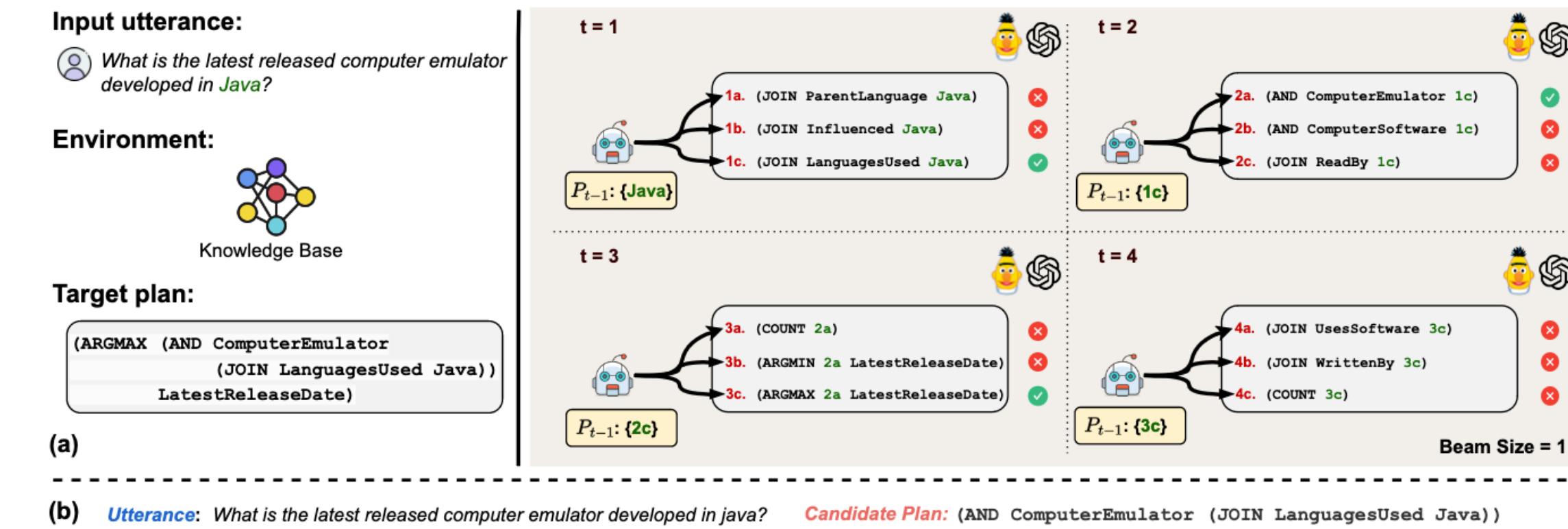
- Semiparametric approach with the generated (Q,P) pairs in nonparametric memory.
- The parametric component is an LLM that uses the demonstration retrieved from the memory
- No fine-tuning - Therefore our method works for any KG 😊

# Stage 3: Bottom-up Grounded Reasoning

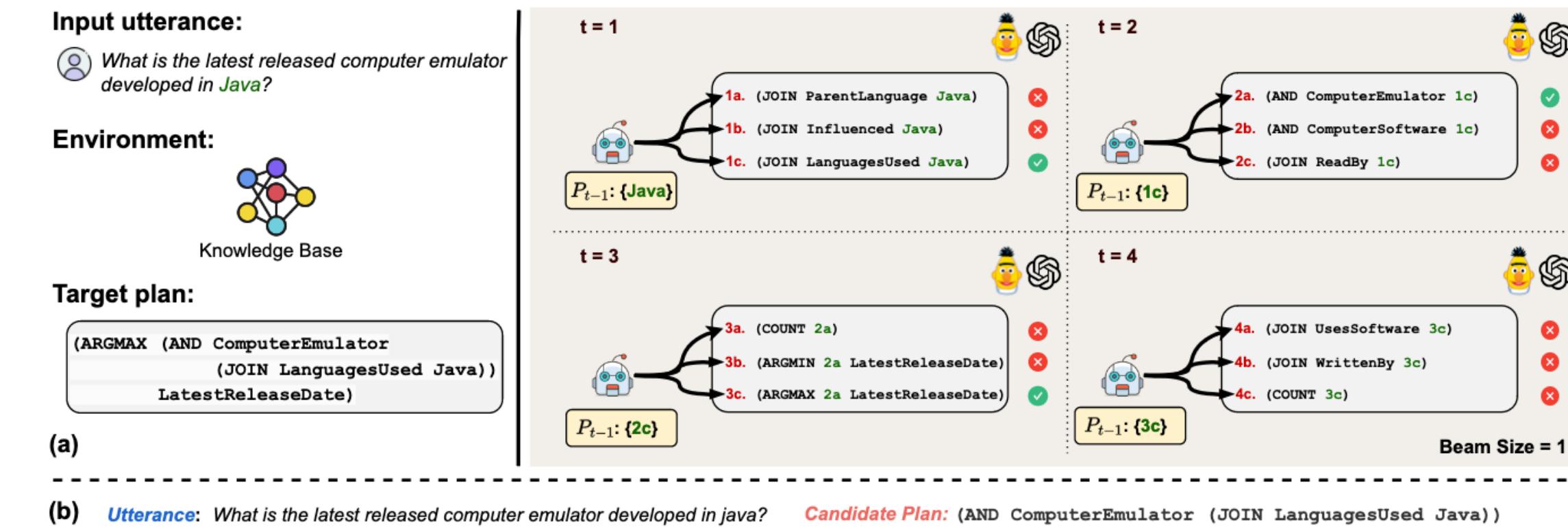
# Stage 3: Bottom-up Grounded Reasoning



# Stage 3: Bottom-up Grounded Reasoning

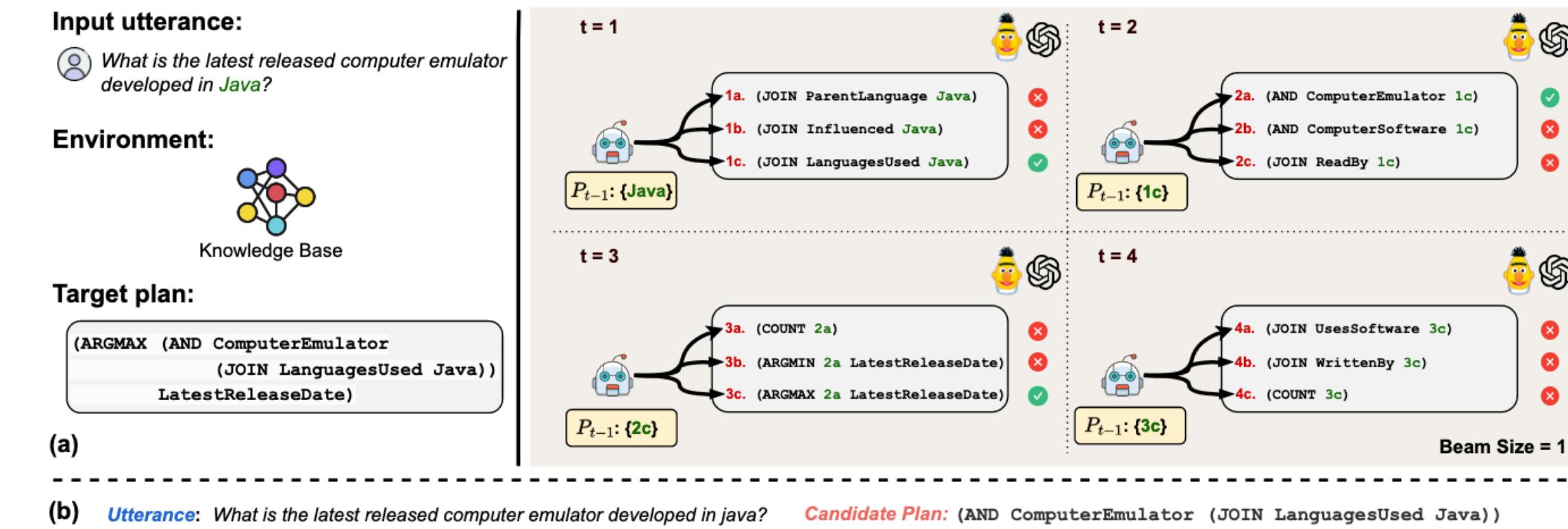


# Stage 3: Bottom-up Grounded Reasoning



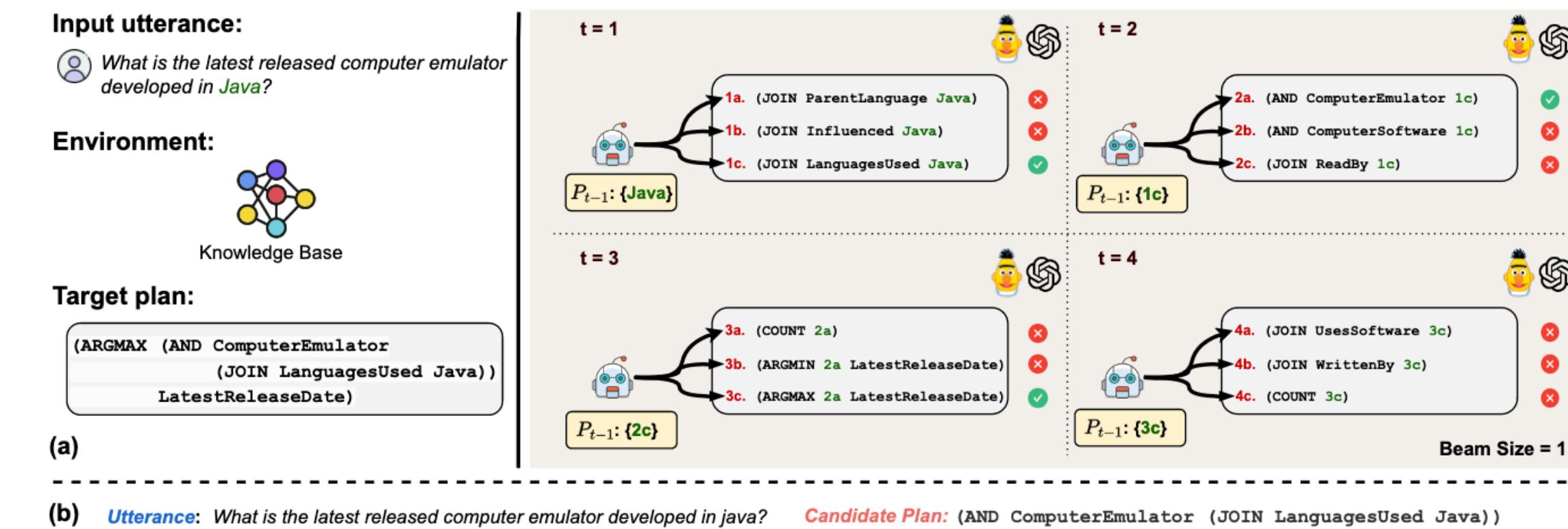
- 2 key changes that improves both speed and accuracy

# Stage 3: Bottom-up Grounded Reasoning



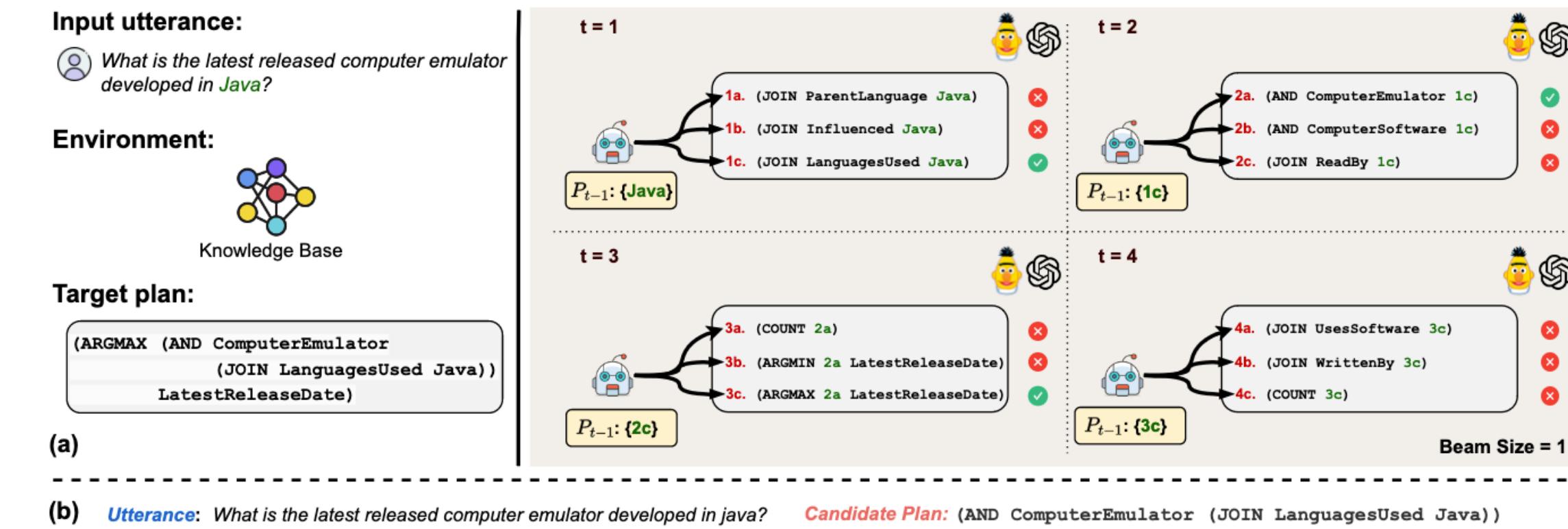
- 2 key changes that improves both speed and accuracy
- Pruning at each step (improves speed by 8.33x)

# Stage 3: Bottom-up Grounded Reasoning



- 2 key changes that improves both speed and accuracy
  - Pruning at each step (improves speed by 8.33x)
  - Inverse-consistency re-ranking (improves accuracy)

# Stage 3: Bottom-up Grounded Reasoning



- 2 key changes that improves both speed and accuracy
  - Pruning at each step (improves speed by 8.33x)
  - Inverse-consistency re-ranking (improves accuracy)

Inverse-consistency is pretty general. You should try out for your generation experiments (esp. w/ smaller models)

# Results

# Datasets and Graphs

- **GrailQA:** 13,231 test questions containing questions up to 4-hops
    - **Freebase KG (Commons subset):** 3.7k relations, 1.5k classes, 32k entities
  - **MetaQA:** 39,093 test questions containing questions up to 3-hops
    - **MoviesKG:** 9 relations, 7 classes, 43k entities
  - **MatKG:** 100 test questions . Unseen KG, 21 relations, 7 classes, 70k entities
- 

## Metrics

- **F1-score**
  - **Answer-EM**
  - **Hits@1**
- 

## Models

- **Open-source:** MPT-7B
- **Closed-source:** GPT-3.5 (sub-sampled experiments)

# **Result #1 Exploration lead to substantial gain in unsupervised setting**

# Result #1 Exploration lead to substantial gain in unsupervised setting

Method	GrailQA (F1)	MetaQA (F1)
--------	-----------------	----------------

# Result #1 Exploration lead to substantial gain in unsupervised setting

Method	GrailQA (F1)	MetaQA (F1)
Zero-shot	18.58	15.43

# Result #1 Exploration lead to substantial gain in unsupervised setting

Method	GrailQA (F1)	MetaQA (F1)
Zero-shot	18.58	15.43
Pangu + exploration (10K)	42.44	54.68

# Result #1 Exploration lead to substantial gain in unsupervised setting

Method	GrailQA (F1)	MetaQA (F1)
Zero-shot	18.58	15.43
Pangu + exploration (10K)	42.44	54.68
BYOKG + exploration (10K)	<b>46.47</b>	<b>75.31</b>

# Results #2: Competitive results with *supervised* setting

# Results #2: Competitive results with *supervised* setting

Method	GrailQA (F1)	MetaQA (F1)
BYOKG + training data (10K)	46.61	82.10
BYOKG + exploration (10K)	46.47	75.31

# Results #3: Better consistency with splits

# Results #3: Better consistency with splits

Method	IID	Compositional	Zero-shot	Overall
<b>BYOKG + training data (10K)</b>	58.29	45.14	41.89	46.61

# Results #3: Better consistency with splits

Method	IID	Compositional	Zero-shot	Overall
BYOKG + training data (10K)	58.29	45.14	41.89	46.61

# Results #3: Better consistency with splits

Method	IID	Compositional	Zero-shot	Overall
<b>BYOKG + training data (10K)</b>	58.29	45.14	41.89	46.61
<b>BYOKG + exploration (10K)</b>	48.91	43.22	46.80	46.47

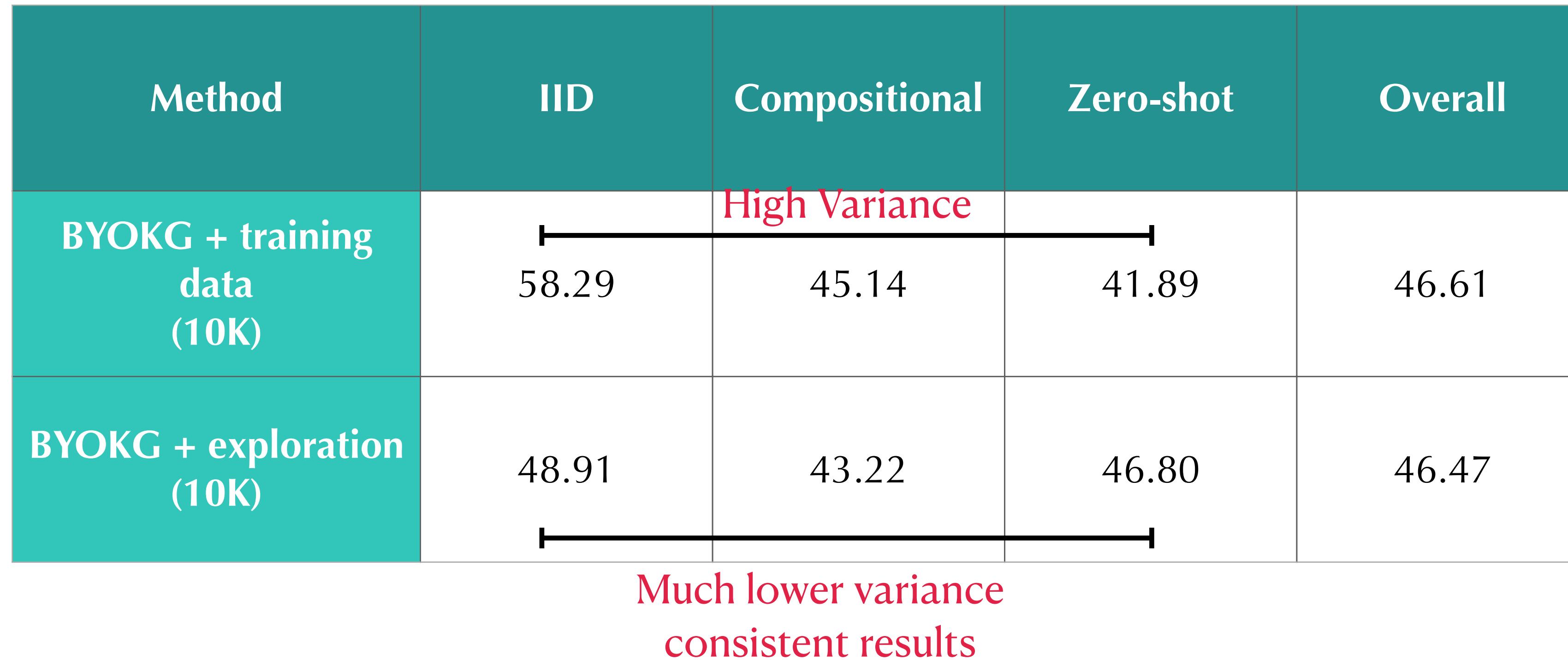
# Results #3: Better consistency with splits

Method	IID	Compositional	Zero-shot	Overall
BYOKG + training data (10K)	58.29	45.14	41.89	46.61
BYOKG + exploration (10K)	48.91	43.22	46.80	46.47

**High Variance**

**Much lower variance  
consistent results**

# Results #3: Better consistency with splits



# Results #4: BYOKG improves with model scale

# Results #4: BYOKG improves with model scale

Method	IID	Compositional	Zero-shot	Overall
Pangu (Codex + Training data)	73.7	64.9	61.1	65.0

# Results #4: BYOKG improves with model scale

Method	IID	Compositional	Zero-shot	Overall
Pangu (Codex + Training data)	73.7	64.9	61.1	65.0
BYOKG (GPT-3.5) + exploration (10K)	<b>73.89</b>	<b>70.33</b>	<b>80.99</b>	<b>75.16</b>

# Results #4: BYOKG improves with model scale

Method	IID	Compositional	Zero-shot	Overall
Pangu (Codex + Training data)	73.7	64.9	61.1	65.0
BYOKG (GPT-3.5) + exploration (10K)	<b>73.89</b>	<b>70.33</b>	<b>80.99</b>	<b>75.16</b>

Performance increases with stronger LLMs and outperforms model using supervised training data

# **Case Study: Material Science KG**

# Case Study: Material Science KG

- ◆ Specialized domain of Material Sciences
- ◆ Totally unseen KG (released after pre-training of MPT models)
- ◆ No training data - we annotate 100 NL question

# Case Study: Material Science KG

- ◆ Specialized domain of Material Sciences
- ◆ Totally unseen KG (released after pre-training of MPT models)
- ◆ No training data - we annotate 100 NL question

Method	Overall
Zero-shot	15.92

# Case Study: Material Science KG

- ◆ Specialized domain of Material Sciences
- ◆ Totally unseen KG (released after pre-training of MPT models)
- ◆ No training data - we annotate 100 NL question

Method	Overall
Zero-shot	15.92
BYOKG + exploration (10K)	<b>62.25</b>

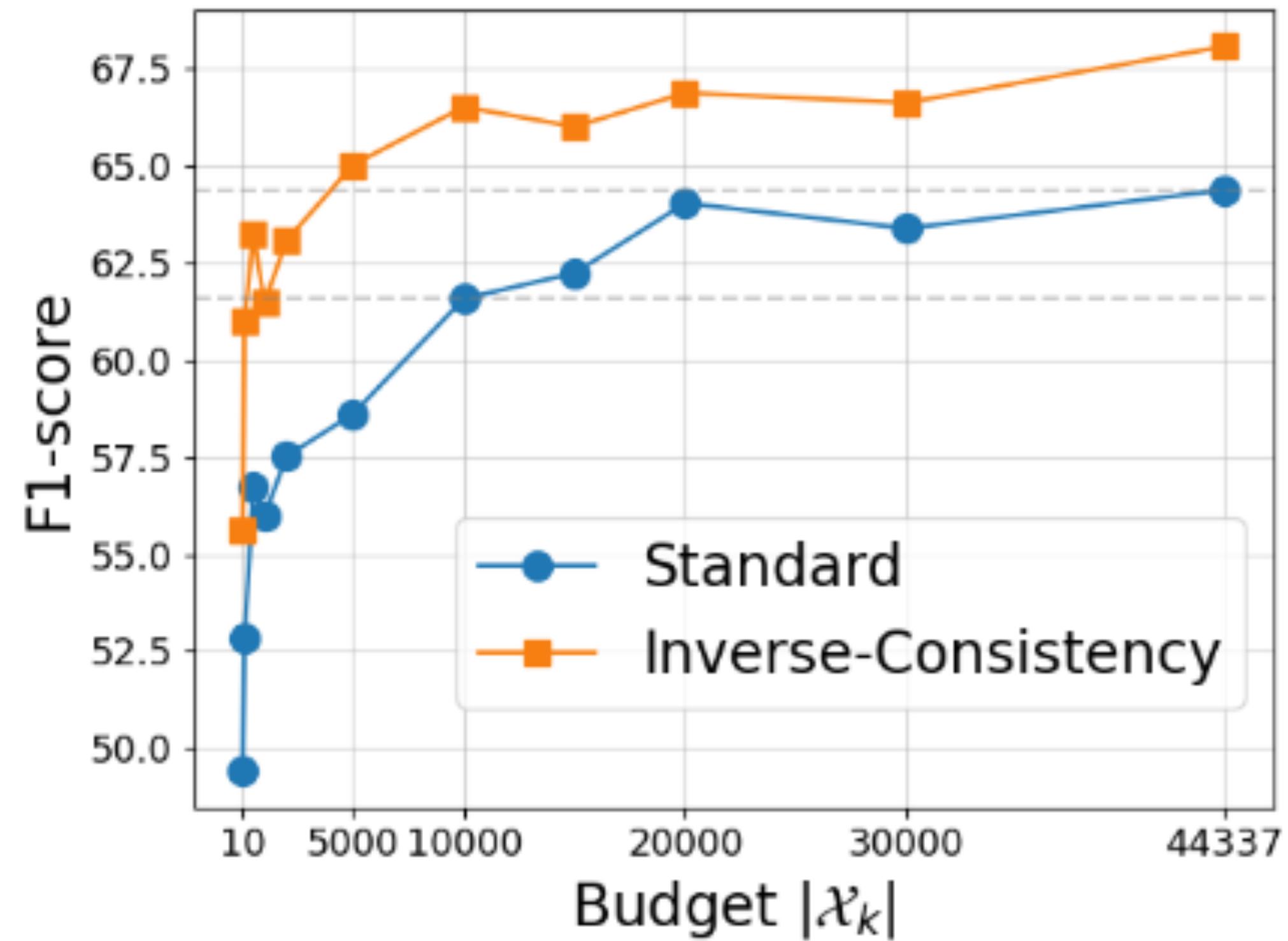
# **Result #5: Inverse Consistency is helpful**

# Result #5: Inverse Consistency is helpful

Metrics	Standard	Inverse-Consistency
ROUGE-1	48.17	<b>52.81</b> ( $\Delta+4.64$ )
BLEU	31.54	<b>38.63</b> ( $\Delta+7.09$ )
BERTscore	87.17	<b>88.33</b> ( $\Delta+1.16$ )
Human Evaluation	47.50	<b>70.00</b> ( $\Delta+22.50$ )

Table 7: **Inverse-Consistency for Question Generation.** Generation quality with inverse-consistency re-ranking compared with standard top-1 predictions from beam search using MPT-7B. Inverse-consistency improves generation quality as measured on both automatic and human evaluation metrics.

# Accuracy v/s Exploration Budget



# **Summary & Future Directions**

# Summary & Future Directions

- ♦ Built a natural language interface for querying KGs that
  - ✓ Works for *Any Knowledge Graph* (irrespective of their schema)
  - ✓ Requires *no training data* (i.e. completely unsupervised)
  - ✓ Ready to be used in *reasonable time* (~1 day)

# Summary & Future Directions

- ♦ Built a natural language interface for querying KGs that
  - ✓ Works for *Any Knowledge Graph* (irrespective of their schema)
  - ✓ Requires *no training data* (i.e. completely unsupervised)
  - ✓ Ready to be used in *reasonable time* (~1 day)
- ♦ Extend this idea to other structured data (databases, web tables, CSVs)
- ♦ Further reduce latency

# Summary & Future Directions

- ♦ Built a natural language interface for querying KGs that
  - ✓ Works for *Any Knowledge Graph* (irrespective of their schema)
  - ✓ Requires *no training data* (i.e. completely unsupervised)
  - ✓ Ready to be used in *reasonable time* (~1 day)
- ♦ Extend this idea to other structured data (databases, web tables, CSVs)
- ♦ Further reduce latency

Thank you!



Code and data: <https://github.com/amazon-science/BYOKG-NAACL24>