Querying graphs with data

Domagoj Vrgoč

University of Edinburgh

Joint work with Leonid Libkin, Wim Martens, Tony Tan, Juan Reutter, Egor Kostylev
What is graph data?
Do people actually query graph data?

Paul Erdős's Bacon number is 3

Paul Erdős and Peter Berg appeared in N Is a Number: A Portrait of Paul Erdős.
Peter Berg and Chris Penn appeared in Corky Romano.
Chris Penn and Kevin Bacon appeared in Footloose.

Erdős–Bacon number - Wikipedia, the free encyclopedia
A person's Erdős–Bacon number is the sum of one's Erdős number—which measures the "collaborative distance" in authoring mathematical papers between ... Scientists - Actors - Others - Table

Erdos-Bacon Numbers
simonsingh.net › Media › Articles › Maths and Science
We were meeting to discuss something else altogether, but he could not resist telling me about the Erdos-Bacon number game, having just matched the world ...
How do people (theorists) query it?

- Base language are Regular path queries (RPQs)
- We’re interested in collaboration connections
- RPQ (:cast ~ :cast)*
How do people (theorists) query it?

- Base language are **Regular path queries (RPQs)**
- We’re interested in collaboration connections
- **RPQ ( :cast ~ :cast )**

Paul Erdős and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are **Regular path queries (RPQs)**
- We’re interested in collaboration connections
- **RPQ**: $(\text{:cast} \rightarrow \text{:cast})^*$

Paul Erdös and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are Regular path queries (RPQs)
- We’re interested in collaboration connections
- RPQ $(\text{:cast} \rightarrow \text{:cast})^*$

Paul Erdös and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are **Regular path queries (RPQs)**
- We’re interested in collaboration connections
- **RPQ (cast∗ cast)**

Paul Erdős and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are Regular path queries (RPQs)
- We’re interested in collaboration connections
- RPQ (:cast ← :cast)*

Paul Erdős and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are Regular path queries (RPQs)
- We’re interested in collaboration connections
- \( \text{RPQ} (\text{:cast} \rightarrow \text{:cast})^* \)

Paul Erdős and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are **Regular path queries (RPQs)**
- We’re interested in collaboration connections
- **RPQ (~ :cast, :cast)**

Paul Erdös and Kevin Bacon have collaborated
How do people (theorists) query it?

- Base language are **Regular path queries (RPQs)**
- We’re interested in collaboration connections
- **RPQ \((:\text{cast} \rightarrow :\text{cast})^*\)**

Paul Erdős and Kevin Bacon have collaborated.
How do people (theorists) query it?

- Base language are Regular path queries (RPQs)
- We're interested in collaboration connections
- RPQ (:cast ← :cast) *

Paul Erdős and Kevin Bacon have collaborated
What is missing?

- RPQs (and many other) disregard the data
  - What if I want people who have a Bacon number?
  - What if I want to find out the actual director?
  - What if I want to see if some movie has two directors?

Our point being:

- Languages that query both data and topology not well understood.
What this work is about:

Design of languages that mix topology and data
What this work is about:

Design of languages that mix topology and data and studying their properties
What did we come up with?

► Several languages based on different principles:
What did we come up with?

- Several languages based on different principles:
  - Regular queries with memory
    - use register automata to specify properties
  - GXPath
    - transfer XPath from trees to graphs
  - TriAL
    - when standard reachability is not enough
What did we come up with?

- Several languages based on different principles:
  - Regular queries with memory
    - use register automata to specify properties
  - Regular queries with data tests
    - restrict register automata for better evaluation complexity
What did we come up with?

- Several languages based on different principles:
  - Regular queries with memory
    - use register automata to specify properties
  - Regular queries with data tests
    - restrict register automata for better evaluation complexity
  - GXPath
    - transfer XPath from trees to graphs
What did we come up with?

Several languages based on different principles:

- **Regular queries with memory**
  - use register automata to specify properties

- **Regular queries with data tests**
  - restrict register automata for better evaluation complexity

- **GXPath**
  - transfer XPath from trees to graphs

- **TriAL**
  - when standard reachability is not enough
Example

- People who have a Bacon number
- \( \langle (: \text{cast} \ - \ : \text{cast})^*: \text{cast}^* \ : \text{cast} [= \text{Kevin Bacon}] \rangle \)
Example

People who have a Bacon number

\[ \langle (\text{:cast}^-\text{:cast})^*\text{:cast}^-\text{:cast}[^{=\text{Kevin Bacon}}]\rangle \]

Sean Penn does
Example

People who have a Bacon number

\[ \langle (:\text{cast}^-:\text{cast})^*:\text{cast}^-:\text{cast}[=\text{Kevin Bacon}] \rangle \]

Sean Penn does
Example

- People who have a Bacon number
- \(\langle (:\text{cast}^- :\text{cast})^*:\text{cast}^- :\text{cast}[^=\text{Kevin Bacon}]\rangle\)

Sean Penn does
Example

- People who have a Bacon number
- \( \langle (:\text{cast}^- :\text{cast})^+ :\text{cast}^- :\text{cast}[=\text{Kevin Bacon}] \rangle \)

Sean Penn does
Example

- People who have a Bacon number
  - $\langle (:\text{cast}^- :\text{cast})^* :\text{cast}^- :\text{cast} = \text{Kevin Bacon} \rangle$

Sean Penn does
But so does Paul Erdős
Example

People who have a Bacon number

\( \langle (:\text{cast}^-:\text{cast})^*:\text{cast}^-:\text{cast}[=\text{Kevin Bacon}] \rangle \)

Sean Penn does
But so does Paul Erdős
Example

- People who have a Bacon number
  - $\langle (\text{cast}^\ast \cdot \text{cast}) \cdot \text{cast}^\ast \cdot \text{cast} \rangle = \text{Kevin Bacon}$

Sean Penn does
But so does Paul Erdős
People who have a Bacon number

\[ \langle (:\text{cast}^- :\text{cast})^*: :\text{cast}^- :\text{cast}[=\text{Kevin Bacon}] \rangle \]

Sean Penn does
But so does Paul Erdős
People who have a Bacon number

\[ \langle (:\text{cast}^{-} :\text{cast})^{*} :\text{cast}^{-} :\text{cast}[=\text{Kevin Bacon}] \rangle \]

Sean Penn does
But so does Paul Erdős
Example

- People who have a Bacon number
  - $\langle (:\text{cast}^- :\text{cast})^*:\text{cast}^- :\text{cast}[=\text{Kevin Bacon}] \rangle$

Sean Penn does
But so does Paul Erdős
Example

- People who have a Bacon number
- \( \langle (:cast^- :cast)^* :cast^- :cast[=Kevin Bacon]\rangle \)

Sean Penn does
But so does Paul Erdös
Example

- People who have a Bacon number
- $\langle (:\text{cast}^- :\text{cast})^* :\text{cast}^- :\text{cast}[=\text{Kevin Bacon}] \rangle$

Sean Penn does
But so does Paul Erdős
Example

- People who have a Bacon number
- \( \langle (:\text{cast}^- :\text{cast})^* :\text{cast}^- :\text{cast} [= \text{Kevin Bacon}] \rangle \)

Sean Penn does
But so does Paul Erdös
Example

- People who have a Bacon number
  - $\langle (:\text{cast}^- :\text{cast})^*: :\text{cast}^- :\text{cast}[=\text{Kevin Bacon}] \rangle$

Sean Penn does
But so does Paul Erdős
Why is this interesting to this community?

- So far I only talked about databases
Why is this interesting to this community?

- So far I only talked about databases
- But to reason about our languages we use:
Why is this interesting to this community?

- So far I only talked about databases
- But to reason about our languages we use:
  - Automata
    - Specification, evaluation, inexpressivity
Why is this interesting to this community?

- So far I only talked about databases
- But to reason about our languages we use:
  - **Automata**
    - Specification, evaluation, inexpressivity
  - **Logics**
    - **FO** usual database yardstick – expressive power, inexpressivity
Why is this interesting to this community?

- So far I only talked about databases
- But to reason about our languages we use:
  - **Automata**
    - Specification, evaluation, inexpressivity
  - **Logics**
    - **FO** usual database yardstick – expressive power, inexpressivity
  - **Games**
    - Separation results
Conclusions

Does talking about graphs with data here make any sense?
Conclusions

Does talking about graphs with data here make any sense?

- Interesting playground for using automata/logic/games
- With data values things are more meaningful
  - But also more challenging (and not less fun)
- You can claim your research has practical applications
Just to end with a picture
Thank you!