Data Path Queries over Embedded Graph Databases

Anthony W. Lin (TU Kaiserslautern & MPI-SWS, Germany)
Joint with Diego Figueira (Univ. Bordeaux, CNRS, Bordeaux INP, France)
Artur Jeż (Univ. of Wroclaw, Poland)

VardiFest'22, Haifa, Israel

Thank you, Moshe!!

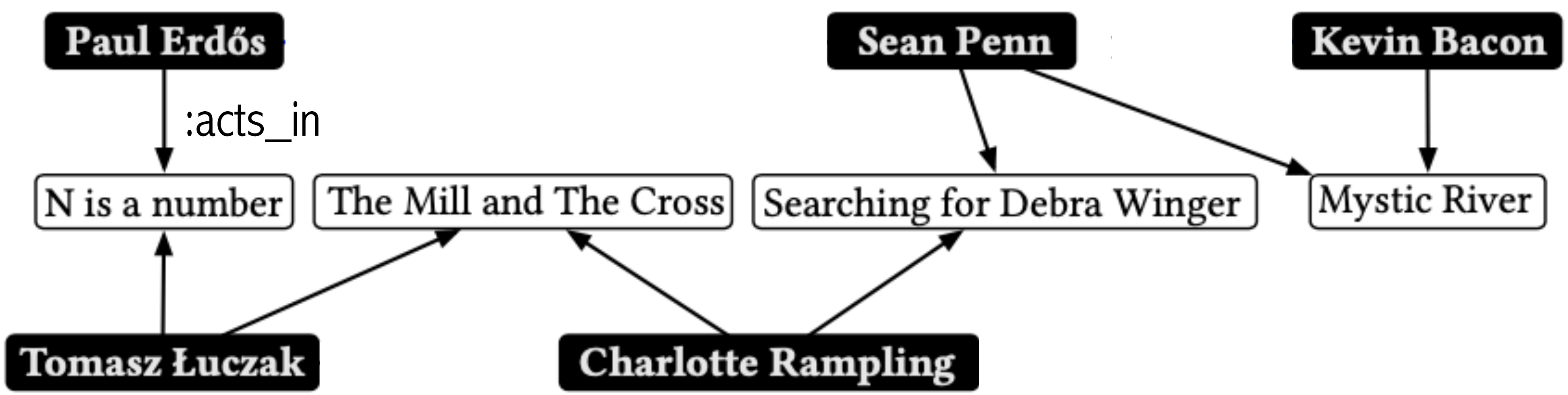
This talk is in honor of Moshe's fundamental contributions in diverse fields especially:

- Database theory (in particular, over graph databases)
- Finite Model Theory
- Automata and Logic
- Boolean satisfiability

The presented result was a modest attempt to learn from Moshe's diversity; it aimed to connect graph databases and SMT

Graph DB: Classic Setting

Output actors that have a finite Bacon number in a movie DB



Regular Path Query (RPQ):

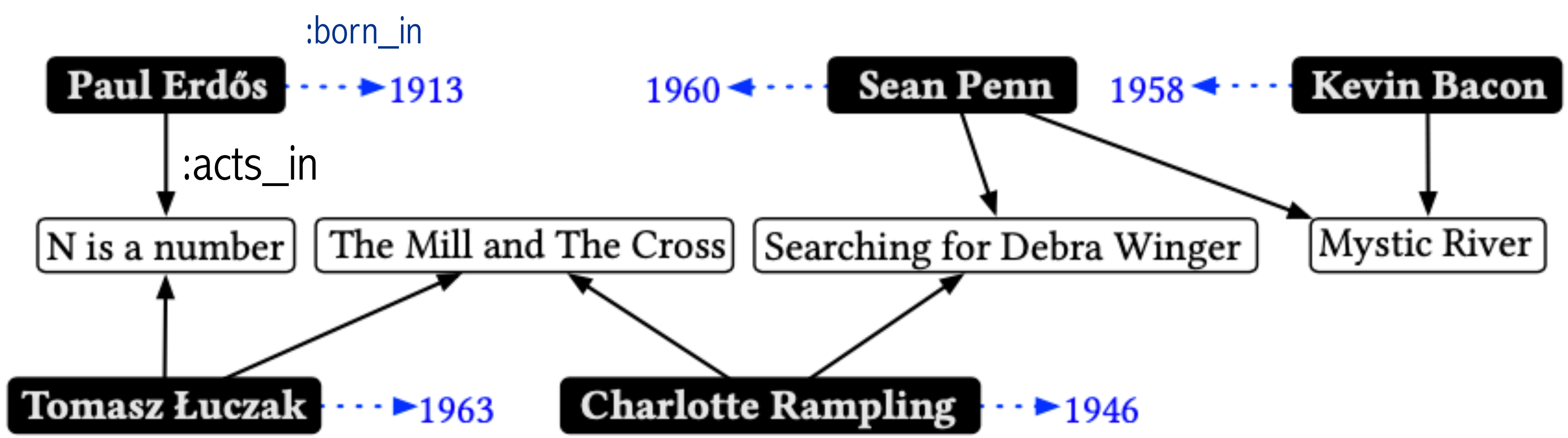
$x \longrightarrow_L$ Bacon, where $L =$ (:acts_in + :acts_in^{-1})*

Desirable data complexity (query L fixed):

NLogspace

„Data“ Querying

Output actors that have a finite Bacon number in a movie DB, whose age is at least 30 years apart from Bacon



Data Queries can get complicated:

1. String data type: similar names along path (small edit distance)
2. Non-linear arithmetics: „nearby“ cities along path (Euclidean distance)

Regular Data Path Queries (RDPQ)

(Libkin, Martens, Vrgoc [early 2010s])

<u>Key idea</u>: data words, register automata (Kaminski&Francez)

Regular Data Path Queries (RDPQ)

(Libkin, Martens, Vrgoc [early 2010s])

Key idea: data words, register automata (Kaminski&Francez)

over $\{$acts_in, acts_in$^{-1}\}$

$\cup\ \mathbb{Z}$

Regular Data Path Queries (RDPQ)

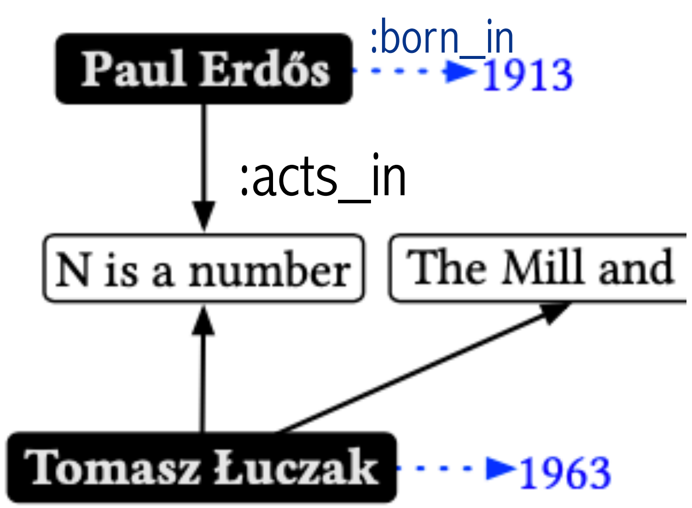
(Libkin, Martens, Vrgoc [early 2010s])

Key idea: data words, register automata (Kaminski&Francez)

over $\{\text{acts\_in}, \text{acts\_in}^{-1}\}$

$\cup\ \mathbb{Z}$

$(1913)(\text{acts\_in})(\text{acts\_in}^{-1})(1963)$



Regular Data Path Queries (RDPQ)

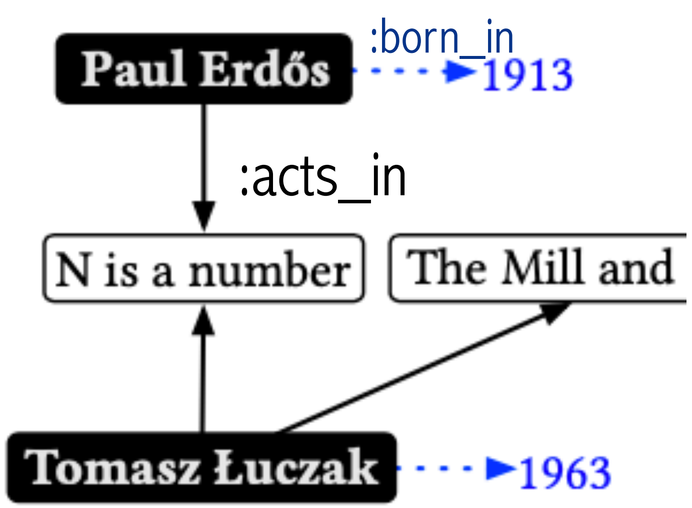
(Libkin, Martens, Vrgoc [early 2010s])

<u>Key idea</u>: data words, register automata (Kaminski&Francez)

over $\{\text{acts\_in}, \text{acts\_in}^{-1}\}$

$\cup\ \mathbb{Z}$

$(1913)(\text{acts\_in})(\text{acts\_in}^{-1})(1963)$



Bacon $\longrightarrow_L$ $Person$, where

$L = x \downarrow (\text{:acts\_in} + \text{: acts\_in}^{-1})^* x^{=}$

Gets actors of equal age

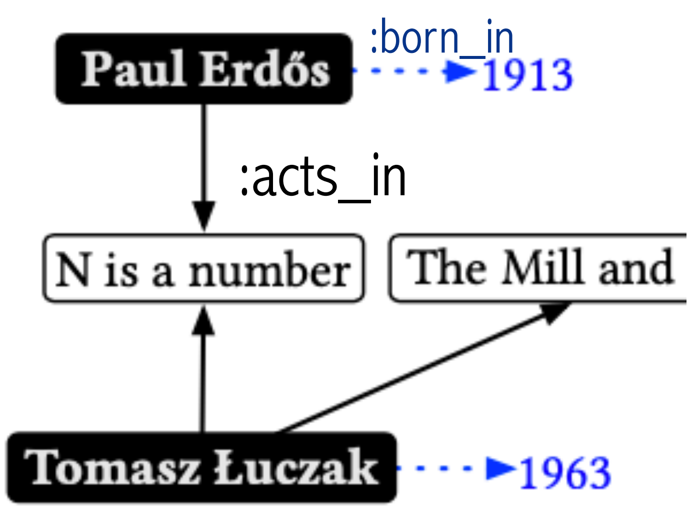
Regular Data Path Queries (RDPQ)

(Libkin, Martens, Vrgoc [early 2010s])

<u>Key idea</u>: data words, register automata (Kaminski&Francez)

over $\{\text{acts\_in}, \text{acts\_in}^{-1}\} \cup \mathbb{Z}$

$(1913)(\text{acts\_in})(\text{acts\_in}^{-1})(1963)$



Bacon $\longrightarrow_L$ $Person$, where

$L = x \downarrow (\text{:acts\_in} + \text{: acts\_in}^{-1})^* x^=$

Gets actors of equal age

Theorem: RDPQ with register automata has NL data complexity.

Regular Data Path Queries (RDPQ)

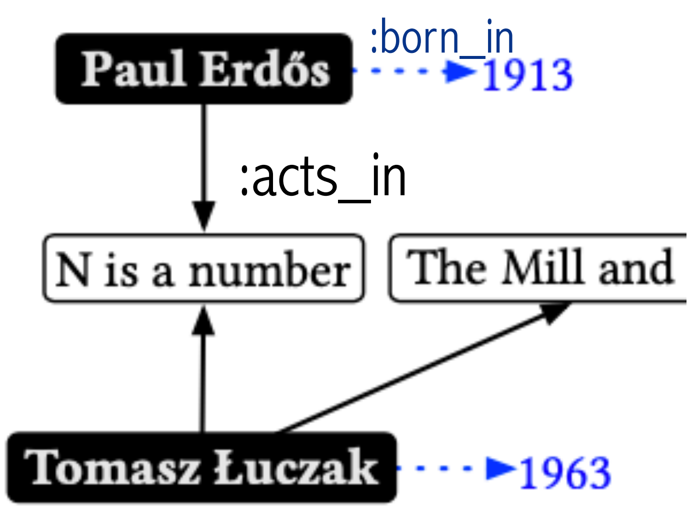
(Libkin, Martens, Vrgoc [early 2010s])

Key idea: data words, register automata (Kaminski&Francez)

over $\{\text{acts\_in}, \text{acts\_in}^{-1}\}$

$\cup\ \mathbb{Z}$

$(1913)(\text{acts\_in})(\text{acts\_in}^{-1})(1963)$



$\text{Bacon} \longrightarrow_L Person$, where

$L = x \downarrow (\text{:acts\_in} + :\ \text{acts\_in}^{-1})^* x^=$

Gets actors of equal age

Theorem: RDPQ with register automata has NL data complexity.

No domain-specific reasoning (e.g. no arithmetics)

Our Main Result

NLogspace data complexity for RDPQ with:

1. Domain-Specific Reasoning (over integer linear arithmetic, theory real closed fields, and various string theories)
2. Generic data graph model

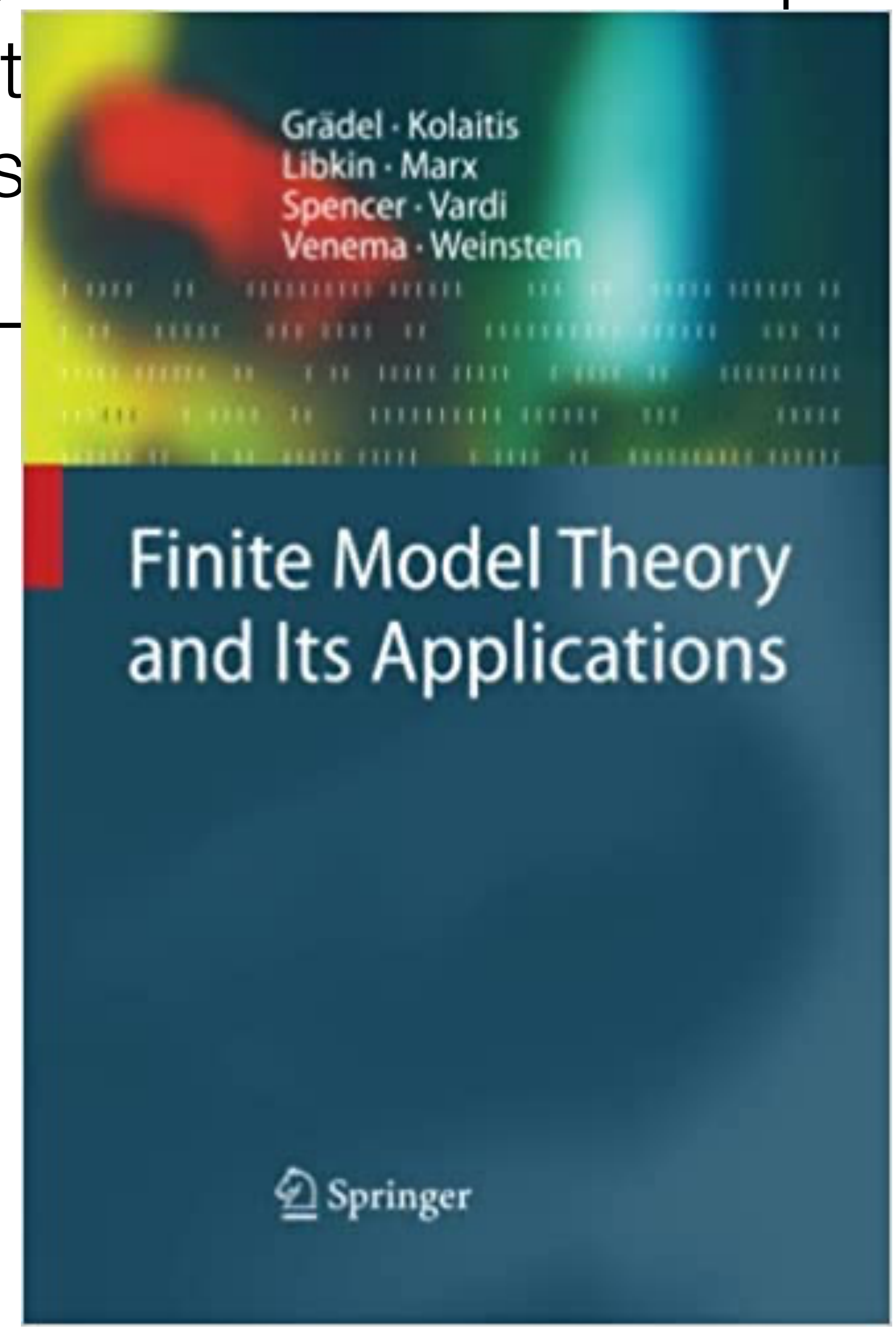
Key ideas:

1. Embedded Finite Model Theory

2. Theory-Aware Register Automata

Our Main Result

NLogspace data complexity for RDPQ with:
1. Domain-Specific Reasoning (over int
   theory real closed fields, and various
2. Generic data graph model



Key ideas:
1. Embedded Finite Model Theory

2. Theory-Aware Register Automata

Key Idea #2: „Theory-Aware“ Register Automata

First approach:

(1) fix an infinite structure $\mathcal{S}$ with a decidable theory

(2) Registers take values and permit operations from $\mathcal{S}$

Problem: undecidable emptiness already for $\mathcal{S} = \langle \mathbb{N}; +1, = \rangle$

Our solution:

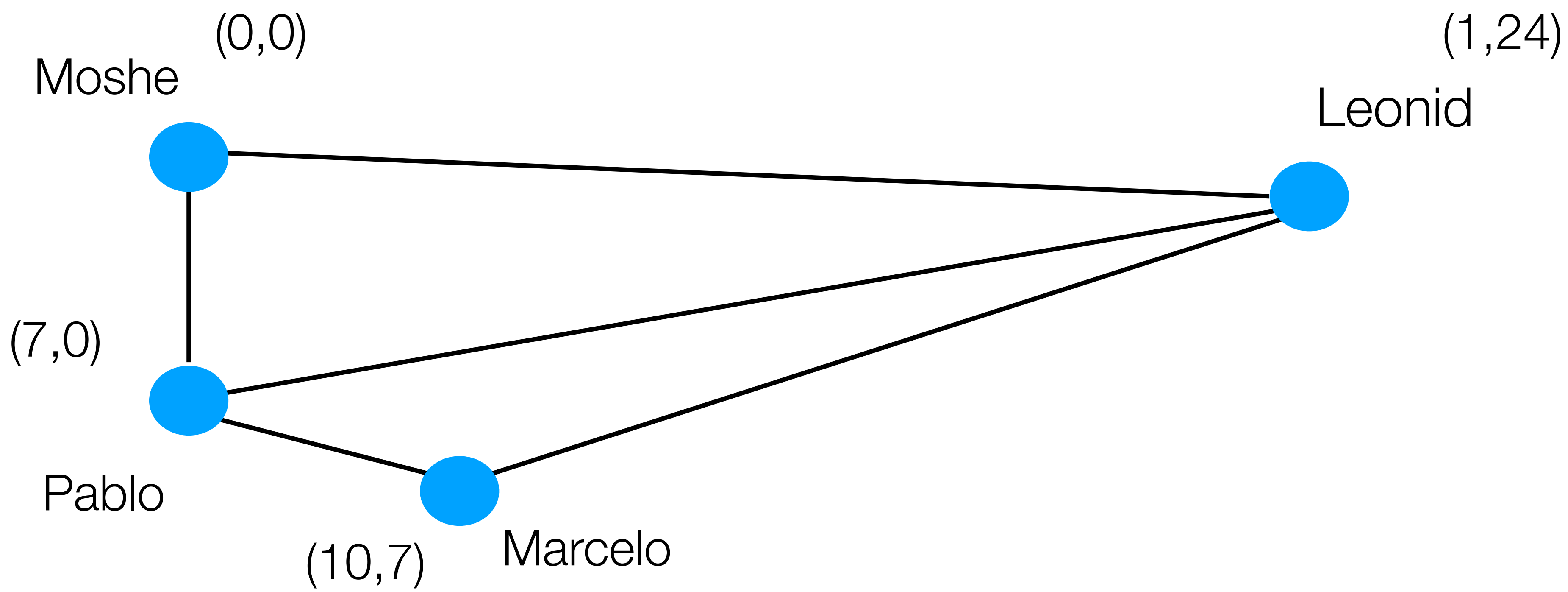
(1) Distinguish between <u>active-domain</u> and <u>general-valued</u> registers

(2) General-valued registers are <u>bounded-rewrite</u>

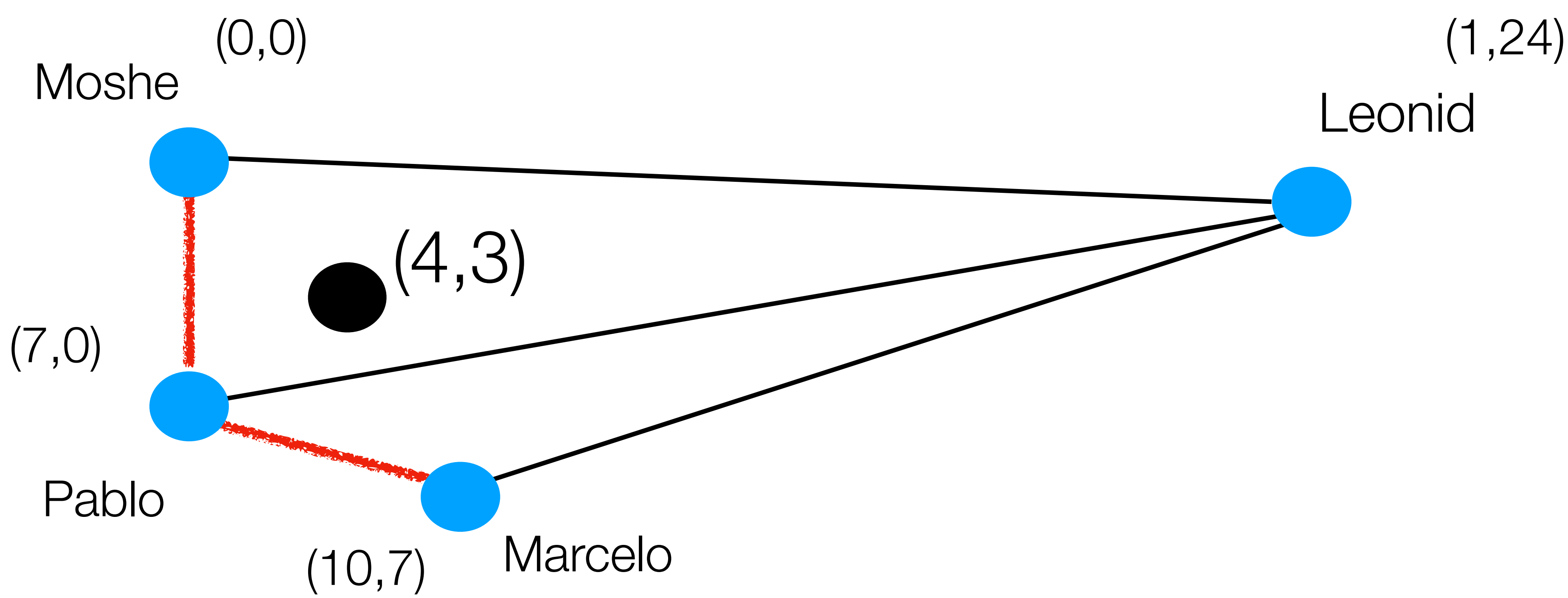
(3) <u>First-order</u> guards

For important theories T (over integers, reals, and strings), we show that T-RDPQ querying still has NL data complexity!

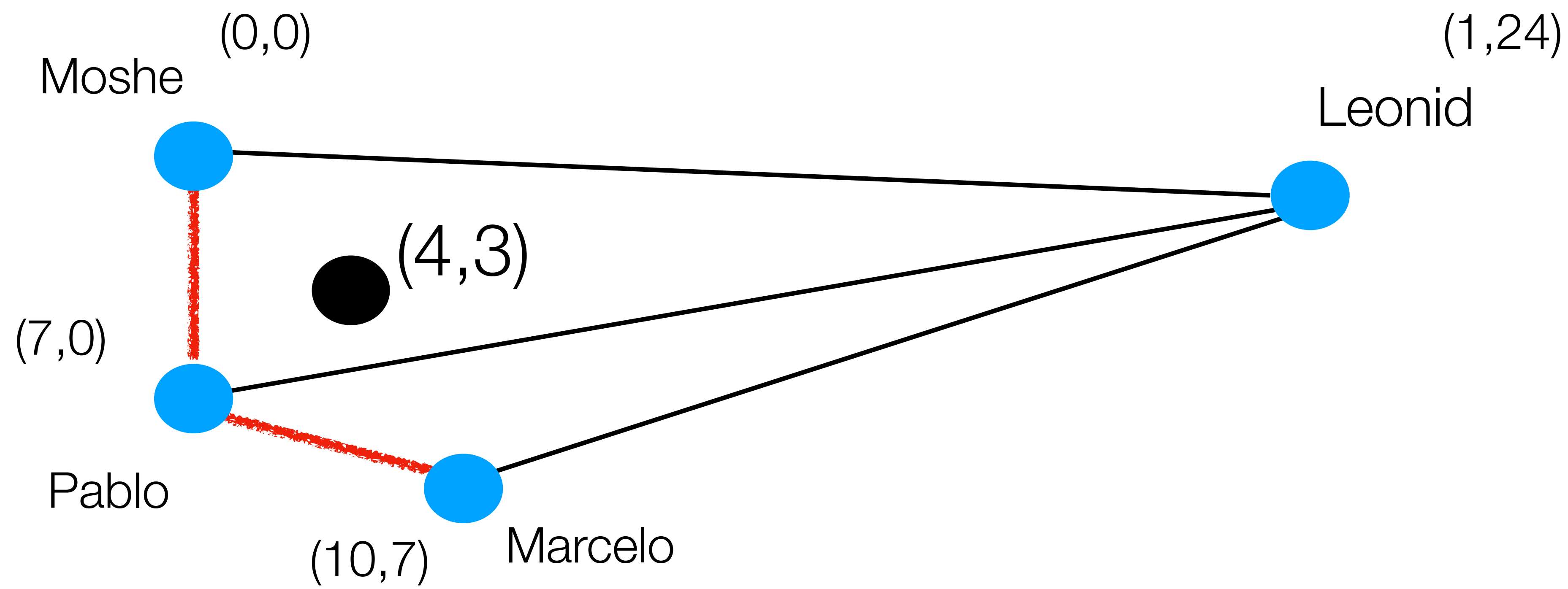
Ex: Path of Coauthors whose „center“ is of distance <= 6



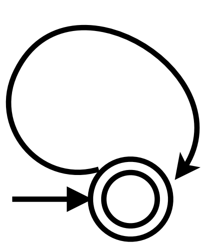
Ex: Path of Coauthors whose „center“ is of distance <= 6



Ex: Path of Coauthors whose „center“ is of distance <= 6



<u>Two unrestricted registers</u>: r_1, r_2



$$\text{coauthors}(curr, next) \wedge$$
$$\exists x, y \in adom \left(\text{xval}(curr, x) \wedge \text{yval}(curr, y) \wedge \sqrt{(x - r_1)^2 + (y - r_2)^2} \leq 6 \right)$$

Theorem (formally)

Theorem:

- RDPQ with $\langle \mathbb{Z}; +, <, 1, 0 \rangle$-RA is NL-complete
- RDPQ with $\langle \mathbb{R}; +, \times, <, 1, 0 \rangle$-RA is NL-complete
- RDPQ with RA over existential positive string equation is NL-complete
- RDPQ with RA over existential automatic structures is NP-hard, but is NL-complete under log-size hypothesis.

Key Technique

Restricted Register Collapse: linear arithmetic, real closed fields

- Each unrestricted register could be effectively replaced by active-domain registers
- Extends the classic notion of Restricted Quantifier Collapse from EFMT

Key Technique

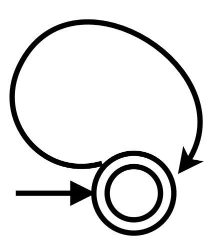
Restricted Register Collapse: linear arithmetic, real closed fields

Each unrestricted register could be effectively replaced by active-domain registers

Extends the classic notion of Restricted Quantifier Collapse from EFMT

Example:

Two unrestricted registers: r_1, r_2



$$\text{coauthors}(curr, next) \land \exists x, y \in adom \left(\text{xval}(curr, x) \land \text{yval}(curr, y) \land \sqrt{(x - r_1)^2 + (y - r_2)^2} \leq 6 \right)$$

Key Technique

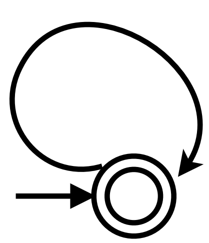
Restricted Register Collapse: linear arithmetic, real closed fields

Each unrestricted register could be effectively replaced by active-domain registers

Extends the classic notion of Restricted Quantifier Collapse from EFMT

Example:

Two unrestricted registers: r_1, r_2



$$\text{coauthors}(curr, next) \wedge \exists x, y \in adom \left(\text{xval}(curr, x) \wedge \text{yval}(curr, y) \wedge \sqrt{(x - r_1)^2 + (y - r_2)^2} \leq 6 \right)$$

To remove r_2, we can rewrite this to an expression in terms of roots of $(x - r_1)^2 + (y - r_2)^2 \leq 36$ treated as univariate r_2-polynomial, for some active-domain values x, y

Future Work

- Query containment for RDPQ and extensions
- NL data complexity for a more expressive query language, e.g., Regular Data Queries (RDQ)?

Thanks!