Symmetric Datalog ≠ Linear Datalog

László Egri\textsuperscript{1},

joint work with Benoît Larose\textsuperscript{2} and Pascal Tesson\textsuperscript{3}

\textsuperscript{1}McGill University
\textsuperscript{2}Concordia University
\textsuperscript{3}Université Laval

UA and CSP, Nashville June 2007
Symmetric Datalog (LE, Larose, Tesson, 2007)
Symmetric Datalog (LE, Larose, Tesson, 2007)
- In logarithmic space (using Reingold, 2005)
Symmetric Datalog (LE, Larose, Tesson, 2007)

- In logarithmic space (using Reingold, 2005)
- Boolean domains + standard complexity assumptions → all CSPs in $L$ are in symmetric Datalog
Symmetric Datalog (LE, Larose, Tesson, 2007)

- In logarithmic space (using Reingold, 2005)
- Boolean domains + standard complexity assumptions $\rightarrow$ all CSPs in L are in symmetric Datalog
- Conjecture: all CSPs in L are in symmetric Datalog
Symmetric Datalog (LE, Larose, Tesson, 2007)
- In logarithmic space (using Reingold, 2005)
- Boolean domains + standard complexity assumptions → all CSPs in $L$ are in symmetric Datalog
- Conjecture: all CSPs in $L$ are in symmetric Datalog

Undirected $st$-connectivity is definable in symmetric Datalog
Symmetric Datalog (LE, Larose, Tesson, 2007)

- In logarithmic space (using Reingold, 2005)
- Boolean domains + standard complexity assumptions → all CSPs in $L$ are in symmetric Datalog
- Conjecture: all CSPs in $L$ are in symmetric Datalog

- Undirected $st$-connectivity is definable in symmetric Datalog
- Directed $st$-connectivity is not definable in symmetric Datalog
Introduction

- Symmetric Datalog (LE, Larose, Tesson, 2007)
  - In logarithmic space (using Reingold, 2005)
  - Boolean domains + standard complexity assumptions → all CSPs in $L$ are in symmetric Datalog
  - Conjecture: all CSPs in $L$ are in symmetric Datalog

- Undirected $st$-connectivity is definable in symmetric Datalog

- Directed $st$-connectivity is not definable in symmetric Datalog
  - Reflexive transitive closure of a binary relation is not definable in symmetric Datalog
Introduction

- Symmetric Datalog (LE, Larose, Tesson, 2007)
  - In logarithmic space (using Reingold, 2005)
  - Boolean domains + standard complexity assumptions $\rightarrow$ all CSPs in $L$ are in symmetric Datalog
  - Conjecture: all CSPs in $L$ are in symmetric Datalog

- Undirected $st$-connectivity is definable in symmetric Datalog
- Directed $st$-connectivity is not definable in symmetric Datalog
  - Reflexive transitive closure of a binary relation is not definable in symmetric Datalog
  - $\neg\text{CSP}(\langle\{0, 1\}; \leq, \{0\}, \{1\}\rangle)$ is not definable in symmetric Datalog
Recap symmetric Datalog through an example
Outline

- Recap symmetric Datalog through an example
- Definitions: free derivation path, the free structure
Outline

- Recap symmetric Datalog through an example
- Definitions: **free derivation path, the free structure**
- Overview of the general proof
Outline

- Recap symmetric Datalog through an example
- Definitions: free derivation path, the free structure
- Overview of the general proof
- The main idea through an example
Datalog and Derivation Path Example

Input Vocabulary:

\[ S^1, T^1, E^2 \]

Linear (Symmetric) Program:

EDB: Extensional Database Predicate
IDB: Intensional Database Predicate

\[
\begin{align*}
I(y) & \leftarrow S(y) \\
I(y) & \leftarrow I(x); E(x, y) \\
(I(x) & \leftarrow I(y); E(x, y)) \\
G & \leftarrow I(y); T(y)
\end{align*}
\]

Input Structure:

\[ S = \{v_5\}, T = \{v_4\} \]

Derivation Path:

\[
\begin{align*}
G & \leftarrow I(y); T(y) \\
I(v_4) & \leftarrow G \\
E(v_3, v_4) & \leftarrow I(v_3) \\
T(v_4) & \leftarrow E(v_3, v_4) \\
E(v_6, v_3) & \leftarrow I(v_6) \\
I(v_5) & \leftarrow E(v_6, v_3) \\
E(v_5, v_6) & \leftarrow I(v_5) \\
S(v_5) & \leftarrow E(v_5, v_6)
\end{align*}
\]

Symmetric Datalog \neq Linear Datalog – p.4/34
The Free Derivation Path

Symmetric Program $\mathcal{D}$:

\begin{align*}
I(y) & \leftarrow S(y) \\
I(y) & \leftarrow I(x); E(x, y) \\
I(x) & \leftarrow I(y); E(x, y) \\
\text{(Rename the vars: } I(y) & \leftarrow I(x); E(y, x)) \\
G & \leftarrow I(y); T(y) \\
\end{align*}

Input Structure:

$E$:

- $E : s \xrightarrow{a} t$
- $S = \{s\}$, $T = \{t\}$

Derivation Path:

- $G \xrightarrow{I(t)} T(t)$
- $I(a) \xrightarrow{E(a, t)}$
- $I(s) \xrightarrow{E(s, a)}$
- $I(a) \xrightarrow{E(a, s)}$
- $I(s) \xrightarrow{E(s, a)}$
- $S(s)$

Free Derivation Path:

- $G \xrightarrow{I(x_1)} T(x_1)$
- $I(x_2) \xrightarrow{E(x_2, x_1)}$
- $I(x_3) \xrightarrow{E(x_3, x_2)}$
- $I(x_4) \xrightarrow{E(x_4, x_3)}$
- $I(x_5) \xrightarrow{E(x_5, x_4)}$
- $S(x_5)$

Symmetric Datalog $\neq$ Linear Datalog – p.5/34
The Free Structure

Free Derivation Path $\mathcal{F}$:

$$\begin{align*}
G & \leftarrow I(x_1) \quad T(x_1) \\
 & \quad I(x_2) \quad E(x_2, x_1) \\
 & \quad I(x_3) \quad E(x_3, x_2) \\
 & \quad I(x_4) \quad E(x_4, x_3) \\
 & \quad I(x_5) \quad E(x_5, x_4) \\
S(x_5) & 
\end{align*}$$

The free structure $\mathcal{F}$ is accepted by $\mathcal{D}$

Free Structure $\mathcal{F}$:

Domain: $\mathcal{F} = \{x_1, x_2, x_3, x_4, x_5\}$

$$E^\mathcal{F}: \quad x_5 \quad x_4 \quad x_3 \quad x_2 \quad x_1$$

$S^\mathcal{F} = \{x_5\}, T^\mathcal{F} = \{x_1\}$
Proof Strategy

- Assume $\exists$ works
Proof Strategy

- Assume $\mathcal{D}$ works
- Input: long enough path
Proof Strategy

- Assume \( \mathcal{D} \) works
- Input: long enough path
- Abstract away, i.e. take the free derivation path \( \mathcal{F} \)
Proof Strategy

- Assume $\mathcal{D}$ works
- Input: long enough path
- Abstract away, i.e. take the free derivation path $\mathcal{F}$
- Using the symmetricity of $\mathcal{D}$, “zig-zag” on $\mathcal{F}$ to create a new free derivation path $\mathcal{F'}$ such that:
Proof Strategy

- Assume $\mathcal{D}$ works
- Input: long enough path
- Abstract away, i.e. take the free derivation path $\mathcal{F}$
- Using the symmetricity of $\mathcal{D}$, “zig-zag” on $\mathcal{F}$ to create a new free derivation path $\mathcal{F}'$ such that:
  - In $\mathcal{F}'$, there is no path from the vertex in $S$ to the vertex in $T$
Proof Strategy

- Assume $\mathcal{D}$ works
- Input: long enough path
- Abstract away, i.e. take the free derivation path $\mathcal{F}$
- Using the symmetricity of $\mathcal{D}$, “zig-zag” on $\mathcal{F}$ to create a new free derivation path $\mathcal{F}'$ such that:
  - In $\mathcal{F}'$, there is no path from the vertex in $S$ to the vertex in $T$
  - $\mathcal{F}'$ is a valid derivation path for $\mathcal{D}$ over the free structure of $\mathcal{F}'$
Proof Strategy

- Assume $\mathcal{D}$ works
- Input: long enough path
- Abstract away, i.e. take the free derivation path $\mathcal{F}$
- Using the symmetricity of $\mathcal{D}$, “zig-zag” on $\mathcal{F}$ to create a new free derivation path $\mathcal{F'}$ such that:
  - In $\mathcal{F'}$, there is no path from the vertex in $S$ to the vertex in $T$
  - $\mathcal{F'}$ is a valid derivation path for $\mathcal{D}$ over the free structure of $\mathcal{F'}$
- Contradiction
Zig-Zag (Simple Example)

Free Derivation Path $\mathcal{F}$:

Zig-zag (mirror) the yellow segment:

1. $I(x_2)$
   $I(x_3)$
   $E(x_3, x_2)$

2. $I(x_2)$
   $I(x_3)$
   $E(x_3, x_2)$

3. $I(x_2)$
   $I(x_3)$
   $E(x_3, x_2)$

$I(y) \leftarrow S(y)$

$I(y) \leftarrow I(x); E(x, y)$

$I(x) \leftarrow I(y); E(x, y)$

$G \leftarrow I(y); T(y)$

*Symmetric Datalog $\not=\text{Linear Datalog} – p.8/34*
Zig-Zag Continued (Simple Example)

Before renaming the variables in mirrored $\mathcal{F}$

Before renaming the variables in mirrored $\mathcal{F}$

Structure $\mathcal{F}$

$E^\mathcal{F}: [x_5, x_4, x_3, x_2, x_1]$

$S^\mathcal{F} = \{x_5\}, T^\mathcal{F} = \{x_1\}$

Structure $\mathcal{F'}$

$E^\mathcal{F'}: [x_7, x_6, x_5, x_4]$

$S^\mathcal{F'} = \{x_7\}, T^\mathcal{F'} = \{x_1\}$

Symmetric Datalog $\neq$ Linear Datalog – p.9/34
Two main complications:
About The General Proof

- Two main complications:
  - There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. **See free structure.**
Two main complications:

- There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. See free structure.
- We disconnect each
Two main complications:

- There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. See free structure.
  - We disconnect each
  - Careful, we do not want to create new paths when we disconnect a path
Two main complications:

- There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. See free structure.
  - We disconnect each
  - Careful, we do not want to create new paths when we disconnect a path
  - A bit technical
Two main complications:

- There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. See free structure.
  - We disconnect each
  - Careful, we do not want to create new paths when we disconnect a path
  - A bit technical
- Arity of the IDBs can be arbitrary (but fixed). See our example program.
About The General Proof

Two main complications:

♦ There could be more than one path from the vertex in $S$ to the vertex in $T$ in a free derivation path. See free structure.
  • We disconnect each
  • Careful, we do not want to create new paths when we disconnect a path
  • A bit technical

♦ Arity of the IDBs can be arbitrary (but fixed). See our example program.
  • We give an intuition how to handle higher arities.
The UV-Path Following Diagram
The UV-Path Following Diagram

Symmetric Datalog ≠ Linear Datalog – p.12/34
The UV-Path Following Diagram
The UV-Path Following Diagram

\[ u \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ v \]

\[ G \]

\[ I_1 \]

\[ I_2 \]

\[ I_3 \]

\[ I_4 \]

\[ I_5 \]

\[ I_1 \]

\[ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \]

Symmetric Datalog ≠ Linear Datalog – p.14/34
Questions
Questions
Questions
Questions
Questions
Questions
Questions
Questions