TECHNISCHE UNIVERSITÄT DRESDEN

Almost Symmetric Linear Arc Monadic Datalog and Transitive Tournaments

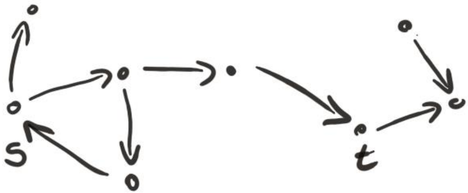
Florian Starke

joint work with Sebastian Meyer

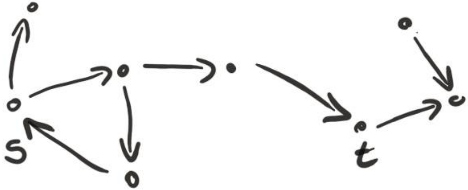
European Research Council
Established by the European Commission

Directed Reachability and Datalog

Directed Reachability
and Datalog



Directed Reachability and Datalog

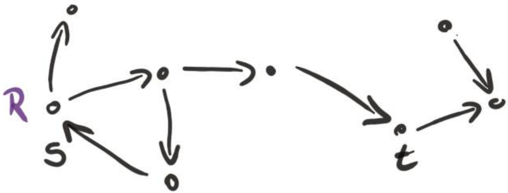


Π_{DR}

$$R(x) :- S(x) \mid \exists y\, R(y) \wedge (y \to x)$$

$$\text{goal} :- R(x) \wedge t(x)$$

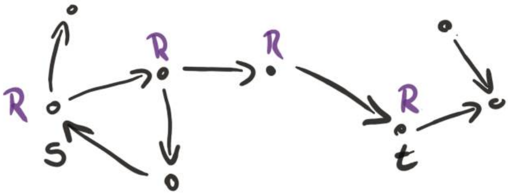
Directed Reachability and Datalog



Π_{DR}

$$R(x) :- S(x) \mid \exists y\, R(y) \wedge (y \to x)$$
$$\text{Goal} :- R(x) \wedge t(x)$$

Directed Reachability and Datalog

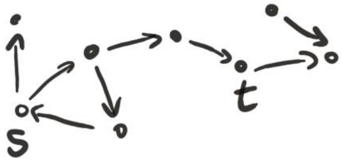


Π_{DR}

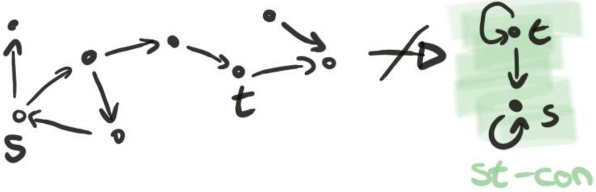
$$R(x) :- S(x) \mid \exists y \, R(y) \wedge (y \to x)$$
$$\text{Goal} :- R(x) \wedge t(x)$$

Directed Reachability and Constraint Satisfaction Problems

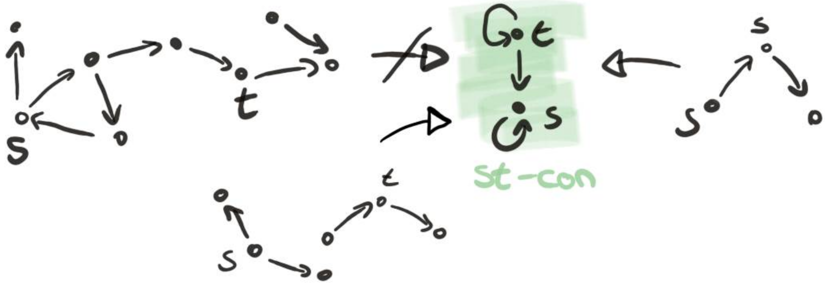
Directed Reachability and Constraint Satisfaction Problems



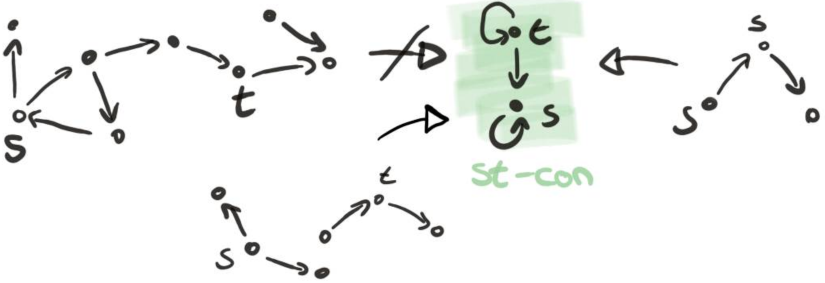
Directed Reachability and Constraint Satisfaction Problems



Directed Reachability and Constraint Satisfaction Problems

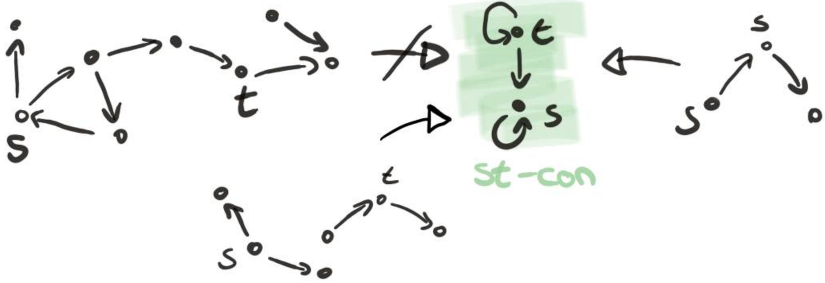


Directed Reachability and Constraint Satisfaction Problems



$\mathrm{CSP}(\text{st-con}) = \{ G \mid G \text{ finite}, G \to \text{st-con} \}$

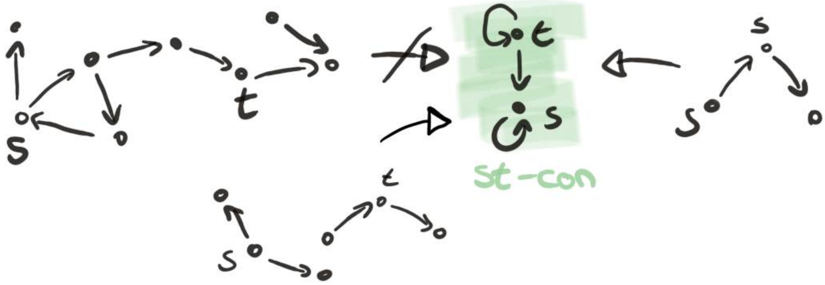
Directed Reachability and Constraint Satisfaction Problems



$$\mathrm{CSP}(\text{st-con}) = \{ G \mid G \text{ finite},\ G \to \text{st-con} \}$$

$$= \{ G \mid G \text{ finite, there is } \underline{\text{no}} \text{ directed path from } s \text{ to } t \text{ in } G \}$$

Directed Reachability and Constraint Satisfaction Problems



$$\mathrm{CSP}(\text{st-con}) = \{ G \mid G \text{ finite}, G \to \text{st-con} \}$$

$$= \{ G \mid G \text{ finite, there is } \underline{\text{no}} \text{ directed path from } s \text{ to } t \text{ in } G \}$$

$$= \{ G \mid G \text{ finite}, \Pi_{DR} \text{ does } \underline{\text{not}} \text{ derive Goal on } G \}$$

Datalog | vs | Constraint Satisfaction Problems

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ

Datalog VS Constraint Satisfaction Problems

DL Program Π with EDBs from τ | CSP ($\mathbb{B}$) template
(finite) τ-structure

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ

CSP ($\mathbb{B}$) template

(finite) τ-structure

$\mathbb{C} \to \mathbb{A}$

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ | CSP (B) template

(finite) τ-structure

$C \to A$

Π derives Goal on C $\Rightarrow$ Π derives Goal on A

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ | CSP (B) template

(finite) τ-structure

$C \to A$

Π derives Goal on C $\Rightarrow$ Π derives Goal on A | $C \to A \to B$

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ	$\mathrm{CSP}(\mathbb{B})$ template (finite) τ-structure

$$C \to A$$

Π derives Goal on C $\Rightarrow$ Π derives Goal on A	$\begin{tikzcd} C \arrow[r] \arrow[rr, bend left] & A \arrow[r] & B \end{tikzcd}$

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ	CSP($\mathbb{B}$) template (finite) τ-structure

$$\mathbb{C} \to \mathbb{A}$$

Π derives Goal on $\mathbb{C}$ $\Rightarrow$ Π derives Goal on $\mathbb{A}$	$\begin{tikzcd} \mathbb{C} \arrow[r] \arrow[rr, bend left] & \mathbb{A} \arrow[r] & \mathbb{B} \end{tikzcd}$ $\mathbb{C} \not\to \mathbb{B} \Rightarrow \mathbb{A} \not\to \mathbb{B}$

Datalog vs Constraint Satisfaction Problems

DL Program Π with EDBs from τ	CSP(B) template; (finite) τ-structure

$C \to A$

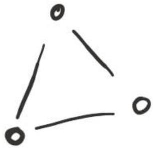
Π derives Goal on C $\Rightarrow$ Π derives Goal on A	$C \to A \to B$; $C \not\to B \Rightarrow A \not\to B$

Π solves CSP(B) iff

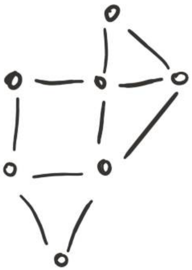
CSP(B) $= \{ A \mid A$ finite, Π does not derive Goal on $A \}$

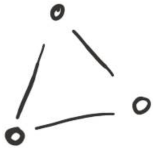
Not all CSPs are solved by Datalog

Not all CSPs are solved by Datalog

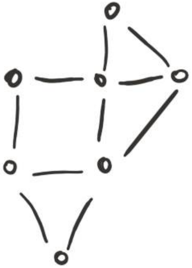


Not all CSPs are solved by Datalog





Not all CSPs are solved by Datalog



Not all CSPs are solved by Datalog

Not all CSPs are solved by Datalog

CSP () = { G | G is 3-colourable }

Not all CSPs are solved by Datalog

CSP () = $\{ G \mid G$ is 3-colourable $\}$

NP-complete

Not all CSPs are solved by Datalog

CSP(△) = { G | G is 3-colourable }

NP-complete, so not solved by Datalog

Polimorphismes

Polimorphismes

$$\mathrm{Pol}(\mathbb{B}) = \{ f \mid n \geq 1,\ f : \mathbb{B}^n \to \mathbb{B} \text{ is a homomorphism} \}$$

Polimorphismes

$\mathrm{Pol}(\mathbb{B}) = \{ f \mid n \geq 1,\ f : \mathbb{B}^n \to \mathbb{B} \text{ is a homomorphism} \}$

$\mathrm{Pol}($  $) \ni \pi_i^{(n)} : (b_1, \dots, b_n) \mapsto b_i$

Polimorphismes

$$\mathrm{Pol}(\mathbb{B}) = \{ f \mid n \geq 1,\ f : \mathbb{B}^n \to \mathbb{B} \text{ is a homomorphism} \}$$

$$\mathrm{Pol}(\triangle) \ni \pi_i^{(n)} : (b_1, \dots, b_n) \mapsto b_i$$

$$\mathrm{Pol}(\text{st-con}) \ni \min$$

$$\|$$

$$\begin{tikzcd} 0 \arrow[loop below] \arrow[r] & 1 \arrow[loop below] \end{tikzcd}$$

$t = \{0\}$

$s = \{1\}$

Polimorphismes

$$\mathrm{Pol}(\mathbb{B}) = \{ f \mid n \geq 1,\ f : \mathbb{B}^n \to \mathbb{B} \text{ is a homomorphism} \}$$

$$\mathrm{Pol}(\triangle) \ni \pi_i^{(n)} : (b_1, \dots, b_n) \mapsto b_i$$

$$\mathrm{Pol}(\text{st-con}) \ni \min$$

$$\begin{array}{l} \min(1 \quad 1 \quad \dots \quad 0) = 0 \\ \qquad\ \ \uparrow \quad \uparrow \qquad\quad \uparrow \qquad \uparrow \\ \min(0 \quad 1 \quad \dots \quad 0) = 0 \end{array}$$

$$\|$$

$$\begin{tikzcd} 0 \arrow[loop below] \arrow[r] & 1 \arrow[loop below] \end{tikzcd}$$

$t = \{0\}$

$s = \{1\}$

Polimorphismes

$$\mathrm{Pol}(\mathbb{B}) = \{ f \mid n \geq 1,\ f : \mathbb{B}^n \to \mathbb{B} \text{ is a homomorphism} \}$$

$$\mathrm{Pol}(\triangle) \ni \pi_i^{(n)} : (b_1, \dots, b_n) \mapsto b_i$$

$$\mathrm{Pol}(\text{st-con}) \ni \min$$
$$\ni \max$$

$$\begin{array}{l} \min(1\ 1\ \dots\ 0) = 0 \\ \quad\ \uparrow\ \ \uparrow \qquad\ \ \uparrow \qquad \uparrow \\ \min(0\ 1\ \dots\ 0) = 0 \end{array}$$

$$\|$$

$$\begin{tikzcd} 0 \arrow[loop below] \arrow[r] & 1 \arrow[loop below] \end{tikzcd}$$

$t = \{0\}$

$s = \{1\}$

Minor Conditions

Minor Conditions

$\min(a,b) = \min(b,a)$

Minor Conditions

$\min(a,b) = \min(b,a)$

$\{a_1, \dots, a_n\} = \{a_1', \dots, a_n'\}$

$\Rightarrow \min(a_1, \dots, a_n) = \min(a_1', \dots, a_n')$

Minor Conditions

$\min(a,b) = \min(b,a)$

$\{a_1, \ldots, a_n\} = \{a_1', \ldots, a_n'\}$

$\Rightarrow \min(a_1, \ldots, a_n) = \min(a_1', \ldots, a_n')$

min is totally symmetric

Minor Conditions

$\min(a,b) = \min(b,a)$

$\{a_1, \dots, a_n\} = \{a_1', \dots, a_n'\}$

$\Rightarrow \min(a_1, \dots, a_n) = \min(a_1', \dots, a_n')$

min is totally symmetric

Theorem

CSP($\mathbb{B}$) is solved by an AM Datalog Program

iff

Pol($\mathbb{B}$) contains totally symmetric Polimorphisms of all arities

Minor Conditions

$\min(a,b) = \min(b,a)$

$\{a_1, \dots, a_n\} = \{a_1', \dots, a_n'\}$

$\Rightarrow \min(a_1, \dots, a_n) = \min(a_1', \dots, a_n')$

min is totally symmetric

Theorem

CSP($\mathbb{B}$) is solved by an AM Datalog Program

IDBs are unary

iff

Pol($\mathbb{B}$) contains totally symmetric Polimorphisms of all arities

Minor Conditions

$\min(a,b) = \min(b,a)$

$\{a_1, \ldots, a_n\} = \{a_1', \ldots, a_n'\}$

$\Rightarrow \min(a_1, \ldots, a_n) = \min(a_1', \ldots, a_n')$

min is totally symmetric

Theorem

$\boxed{\text{CSP}(\mathbb{B}) \text{ is solved by an AM Datalog Program iff Pol}(\mathbb{B}) \text{ contains totally symmetric polimorphisms of all arities}}$

at most one EDB in body

IDBs are uninary

Minor Conditions

Minor Conditions

Definition

A minor condition is a set of identities of the form
$f(x_1, \dots, x_n) = g(y_1, \dots, y_n)$

Minor Conditions

Definition

A minor condition is a set of identities of the form
$f(x_1, \dots, x_n) = g(y_1, \dots, y_n)$

$f(x, f(y,z)) = f(f(x,y), z)$ is <u>not</u> a minor condition

Minor Conditions

Definition

A minor condition is a set of identities of the form

$$f(x_1, \dots, x_n) = g(y_1, \dots, y_n)$$

$f(x, f(y,z)) = f(f(x,y), z)$ is **not** a minor condition

$$f(a_1, a_2, \dots, a_n) = f(a_2, a_1, \dots, a_n) = f(a_2, a_3, \dots, a_1)$$

$f(a_1, a_1, a_3, \dots, a_n) = f(a_1, a_3, a_3, \dots, a_n)$ is a minor condition

Minor Conditions

Definition

A minor condition is a set of identities of the form
$f(x_1, \dots, x_n) = g(y_1, \dots, y_n)$

$f(x, f(y, z)) = f(f(x, y), z)$ is **not** a minor condition

$f(a_1, a_2, \dots, a_n) = f(a_2, a_1, \dots, a_n) = f(a_2, a_3, \dots, a_1)$

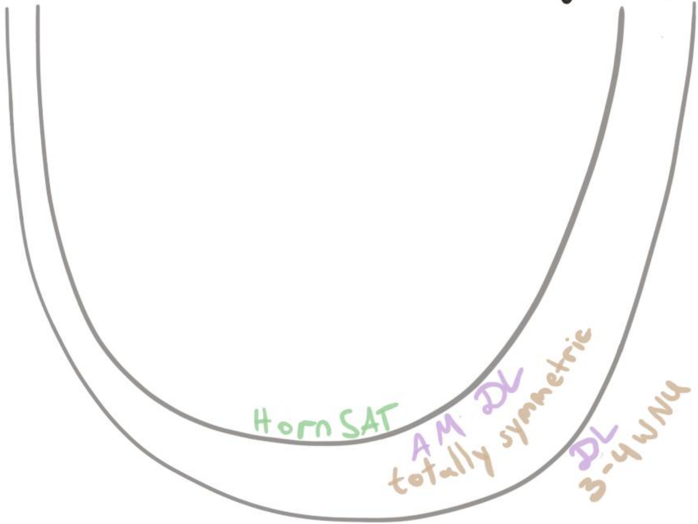
$f(a_1, a_1, a_3, \dots, a_n) = f(a_1, a_3, a_3, \dots, a_n)$ is a minor condition

$\mathbb{A} \leq \mathbb{B}$ if all minor conditions satisfied by $\mathrm{Pol}(\mathbb{A})$ are satisfied by $\mathrm{Pol}(\mathbb{B})$

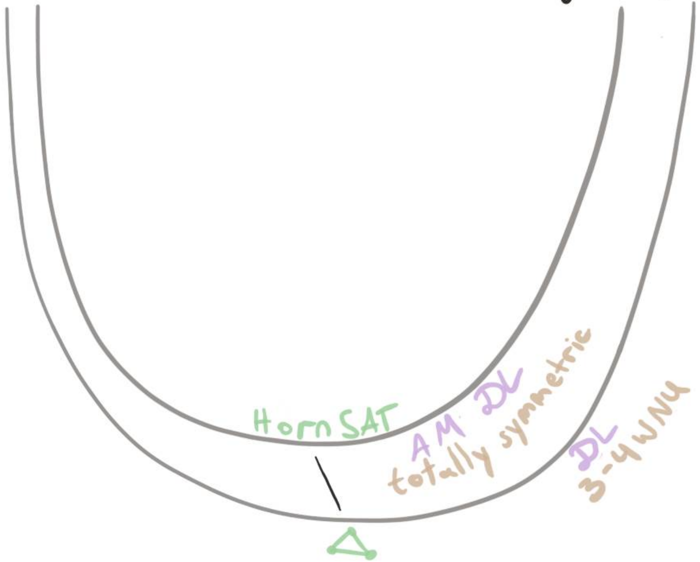
Minor Conditions & Datalog Fragments

Minor Conditions & Datalog Fragments

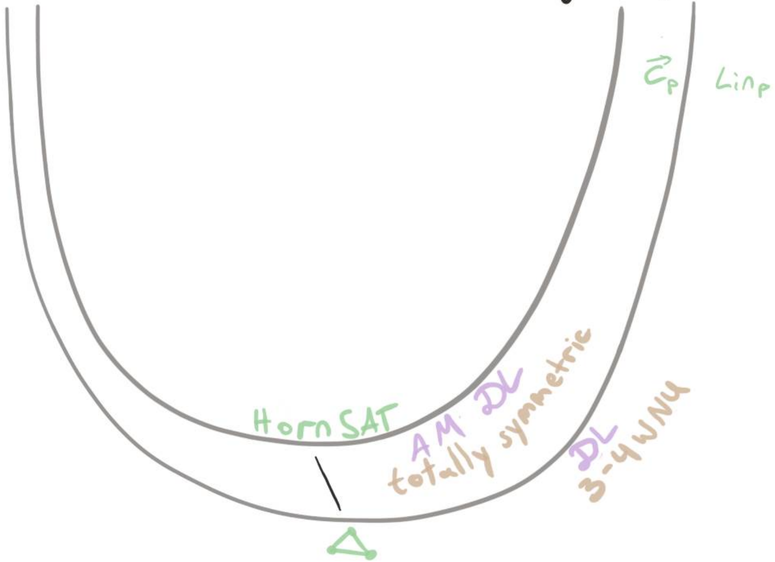
Minor Conditions & Datalog Fragments



Minor Conditions & Datalog Fragments

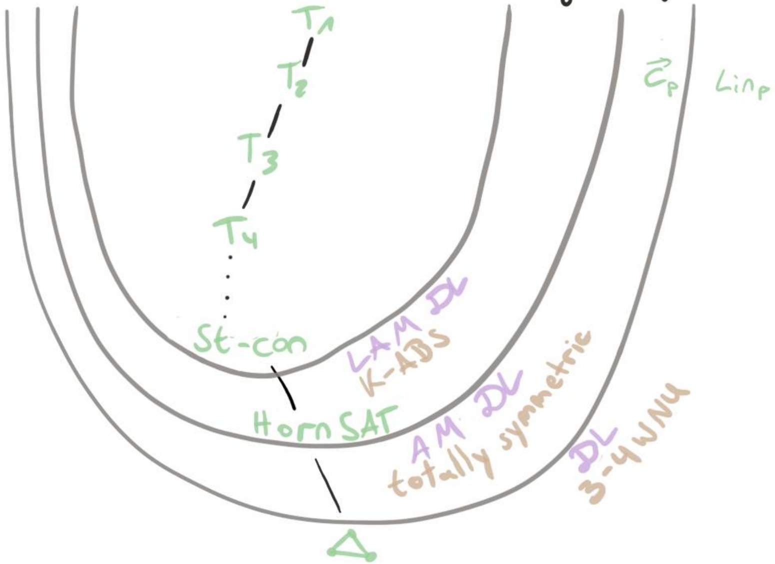


Minor Conditions & Datalog Fragments

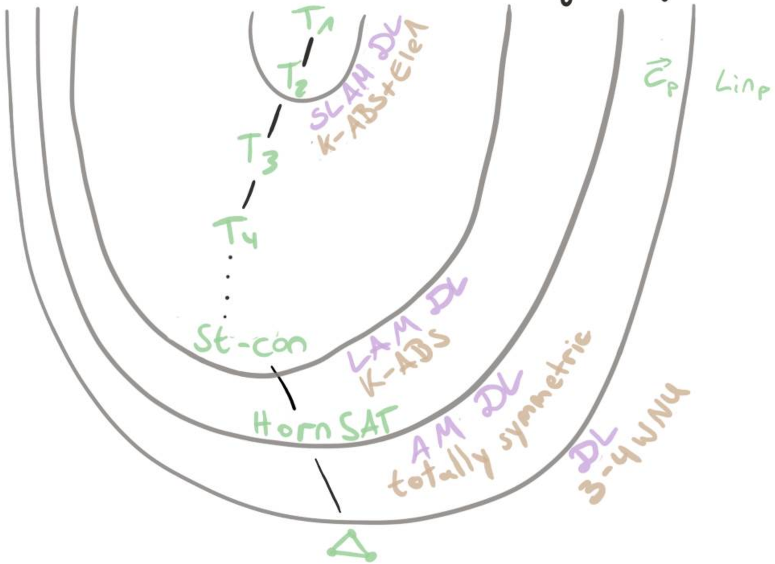


Minor Conditions & Datalog Fragments

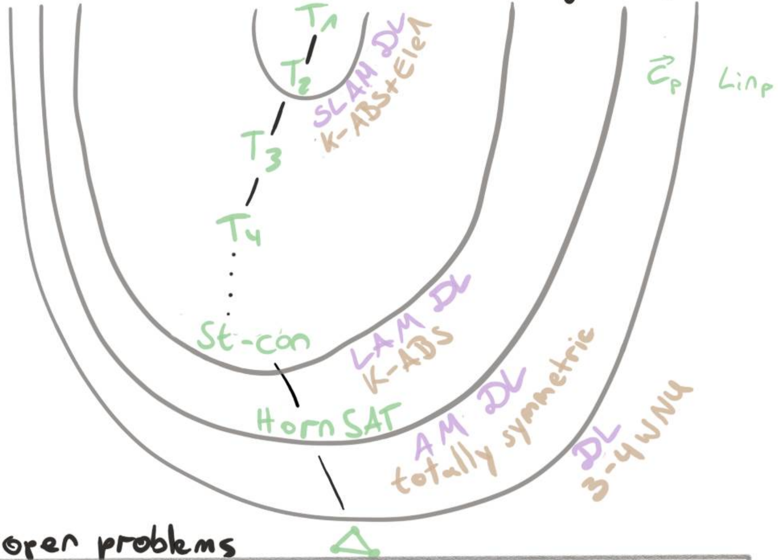
Minor Conditions & Datalog Fragments



Minor Conditions & Datalog Fragments



Minor Conditions & Datalog Fragments



Big open problems

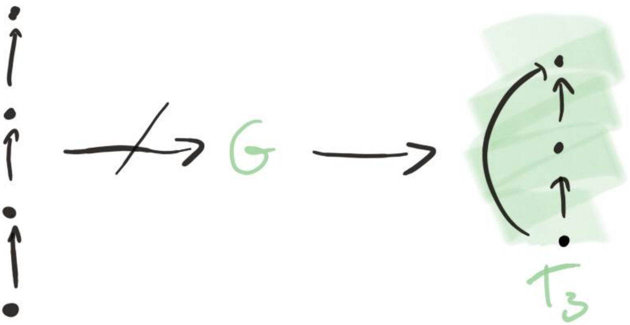
$\boxed{\text{find minor conditions for } l\text{ DL and } sl \text{ DL}}$

CSPs of Transitive Tournaments

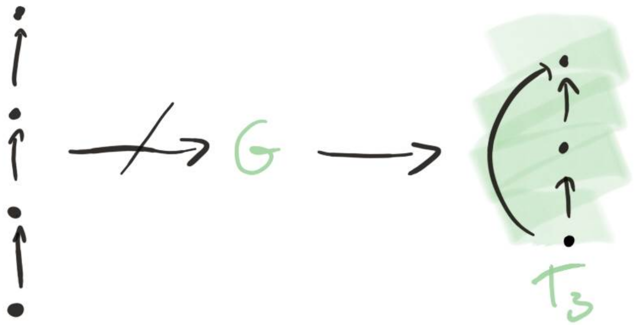
CSPs of Transitive Tournaments

CSPs of Transitive Tournaments

CSPs of Transitive Tournaments



CSPs of Transitive Tournaments

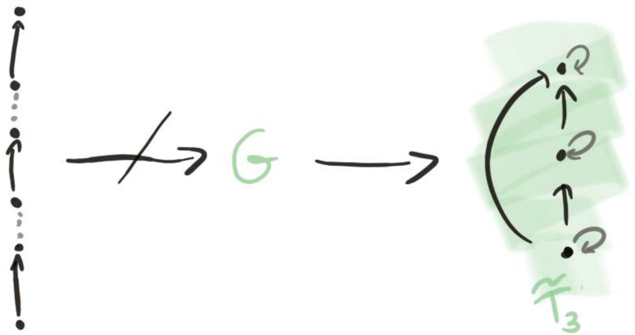


Goal: $\neg \exists x_0\, x_1\, x_2\, x_3 \quad x_0 \to x_1 \to x_2 \to x_3$

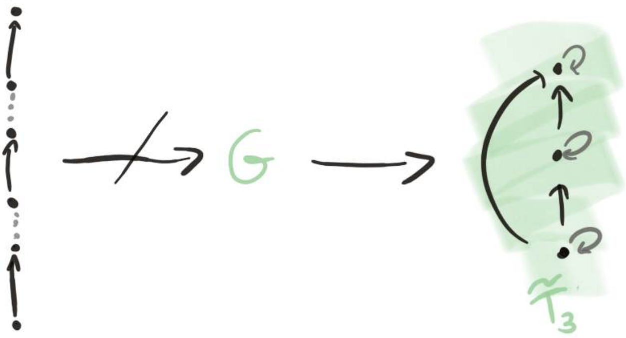
CSPs of Transitive Tournaments

CSPs of Transitive Tournaments

CSPs of Transitive Tournaments



CSPs of Transitive Tournaments

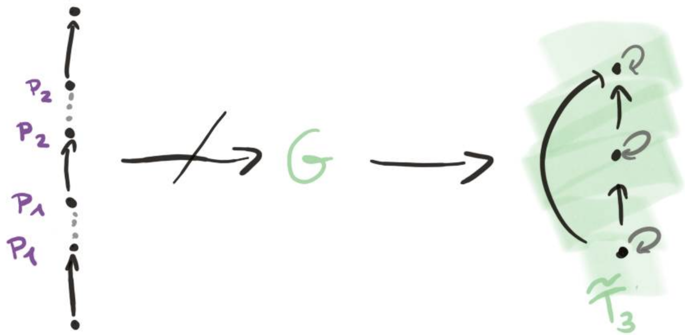


$P_1(x) :- \exists y \; y \to x$

$P_2(x) :- \exists y \; P_1(y) \wedge y \to x$

$\text{Goal} :- \exists x, y \; P_2(y) \wedge y \to x$

CSPs of Transitive Tournaments

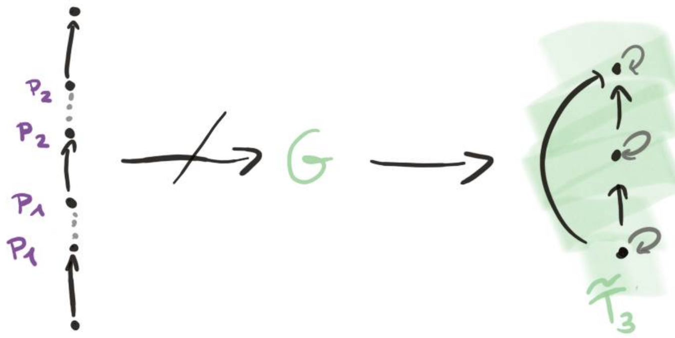


$P_1(x) :- \exists y \; y \to x$

$P_2(x) :- \exists y \; P_1(y) \wedge y \to x$

Goal $:- \exists x, y \; P_2(y) \wedge y \to x$

CSPs of Transitive Tournaments



$P_1(x) :- \exists y \; y \to x \mid \exists y \; P_1(y) \land y \to x \mid \exists y \; P_1(y) \land y \leftarrow x$

$P_2(x) :- \exists y \; P_1(y) \land y \to x \mid \exists y \; P_2(y) \land y \to x \mid \exists y \; P_2(y) \land y \leftarrow x$

$\text{Goal} :- \exists x, y \; P_2(y) \land y \to x$

n-almost symmetric linear arc monadic Datalog

n-almost symmetric linear arc monadic Datalog

Definition

Π is a S_n LAM DL program if

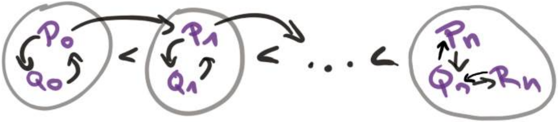
- Π is a LAM DL program

n-almost symmetric linear arc monadic Datalog

Definition

Π is a S_n LAM DL program if

- Π is a LAM DL program
- there is an order on the IDBs with

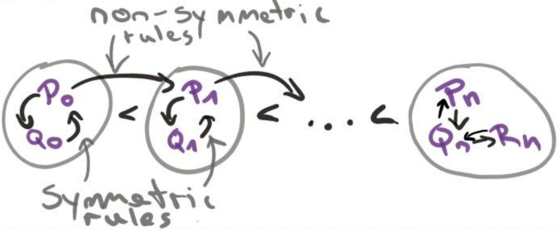


n-almost symmetric linear arc monadic Datalog

Definition

Π is a S_nLAM DL program if

- Π is a LAM DL program
- there is an order on the IDBs with

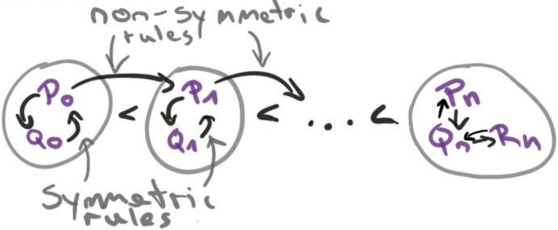


n-almost symmetric linear arc monadic Datalog

Definition

Π is a S_n LAM DL program if

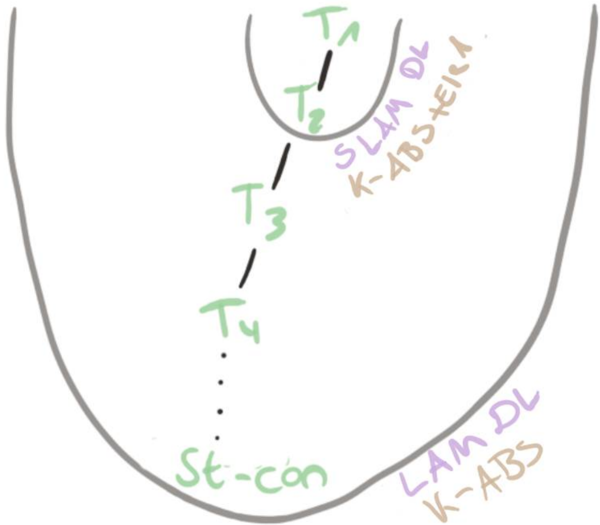
- Π is a LAM DL program
- there is an order on the IDBs with



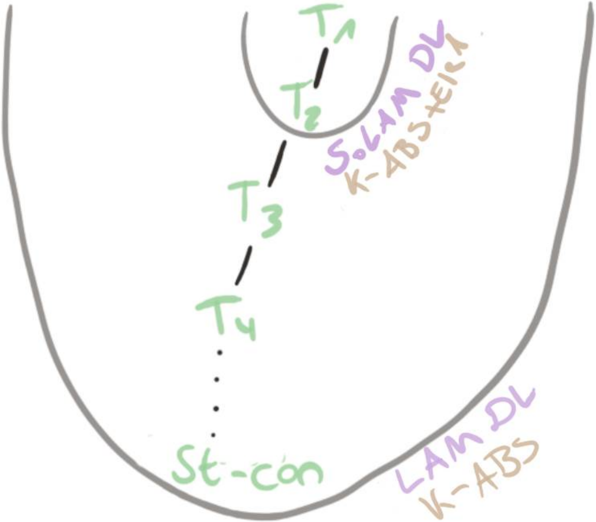
every derivation of Π applies non-symmetric rules at most n times

Results

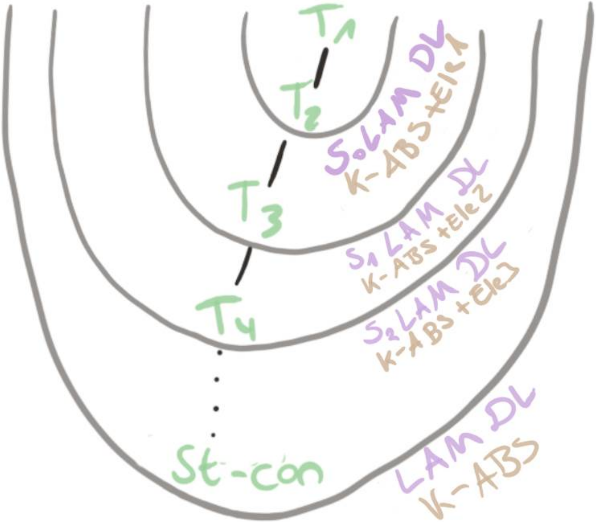
Results



Results



Results



T1
T2
T3
T4
St-con
S0LAM DL
K-ABS+Ele1
S1LAM DL
K-ABS+Ele2
S2LAM DL
K-ABS+Ele3
LAM DL
K-ABS

Results

Thank you for your attention