Automata-theoretic and Datalog-based solutions of MSO Evaluation Problems over Structures of bounded-treewidth

Labrini Kalantzi
University of Athens
(joint work with E. Foustoucos)

2nd Workshop on Graph Decompositions
Theoretical, Algorithmic and Logical Aspects

October 18-22, 2010
CIRM, Luminy, Marseille

Subject & Motivation

- The **MSO evaluation problem** :
  - an MSO *formula*: *defines a property* that characterizes tuples of nodes/subsets of nodes
  - find such tuples that satisfy this property in a given relational structure
- Monadic Second Order logic (**MSO**) : rather expressive language, e.g. defines 3-colorability of graphs
- **Linear algorithms** over structures of bounded treewidth (Courcelle's Theorem)
- We propose a **direct automata-theoretic reduction**
- We propose evaluation of whole MSO via **datalog** (Gottlob ,Pichler, Wei: a datalog solution for restricted class of unary MSO queries)

Talk Outline

The problem

- **Input:** a relational τ-structure $\mathcal{A}$, $\tau = \{R_1, \ldots, R_\ell\}$;
  a (binary, ordered) tree-decomposition $\mathcal{D}$ of $\mathcal{A}$ of width w;
  an MSO[τ] formula $\phi(X_1, \ldots, X_k)$
  Output: Set $\phi(\mathcal{A})$ of satisfying assignments of ϕ over $\mathcal{A}$

We define assignment automata that produce $\phi(\mathcal{A})$ &
then use them to provide a datalog solution for the computation of $\phi(\mathcal{A})$.

Basic facts about our automata: w-*dec-Assign*$_\phi$

A w-*decomposition (assignment) automaton* for $\phi(X_1, \ldots, X_k)$

- *non-deterministic bottom-up tree automaton*, runs over trees encoding a $(\mathcal{A}, \mathcal{D})$ & a *decoding mechanism*
- *special form of state set*

$$Q \times W^k;$$

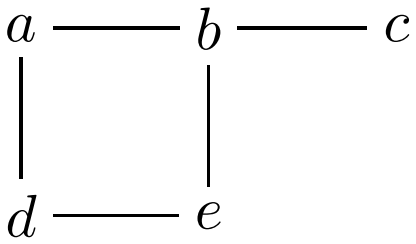
$W = \{1, \ldots, w+1\}$: set of bag positions of $\mathcal{D}$
Thus runs

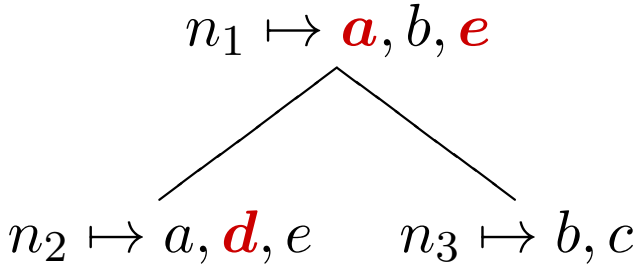
$$\rho = \sigma; \varepsilon.$$

- w-*dec-Assign*$_\phi$ to *produces* $\phi(\mathcal{A})$ as *part of its successful runs*, in an *encoded form* $\sim$ automaton accepting $\mathcal{D}; \varepsilon$ or $(\mathcal{D}, \mathcal{A}); \varepsilon$
- the decoding is performed w.r.t. $\mathcal{D}$

An example of a run producing an assignment of $\phi(X)$

Graph $\mathcal{G}$, ordered binary tree-decomposition $\mathcal{D}$ ($W = \{1, 2, 3\}$)





w-dec-Assign$_\phi$ computes $B = \{a, e, d\} \in \phi(\mathcal{G})$ via a successful run

nodes:	n_1		n_2		n_3
a run $\rho = \sigma; \varepsilon$:	$(q_1, \{1, 3\})$		$(q_2, \{2\})$		$(q_3, \emptyset)$
decoding ε:	$\downarrow \pi_{\{1,3\}}(a, b, e)$		$\downarrow \pi_{\{2\}}(a, b, e)$		$\downarrow \pi_{\emptyset}(b, c)$
B_ε :	$\{a, e\}$	$\bigcup$	$\{d\}$	$\bigcup$	$\emptyset$

Automata select elements via their top-most occurrence in $\mathcal{D}$

Decoding $\varepsilon : T \mapsto W^k$ to $\overline{B}_\varepsilon \in (\mathcal{P}(A))^k$ w.r.t. $\mathcal{D}$

The decoding procedure w.r.t. $\mathcal{D} = (\mathcal{T}, (\overline{a}_n)_{n \in T})$

- At each node n, compute $B_{\varepsilon,n}$

$$\begin{array}{rccc} \varepsilon(n) = & (I^1 & \dots & I^k) \\ & \downarrow_{\mathcal{D}} & \dots & \downarrow_{\mathcal{D}} \\ B_{\varepsilon,n} = & (\pi_{I^1}\overline{a}_n & \dots & \pi_{I^k}\overline{a}_n) \end{array}$$

- Perform component-wise union

$$\overline{B}_\varepsilon = \bigcup_{n \in T} B_{\varepsilon,n}$$

Set of computed assignments

$$w\text{-}dec\text{-}Assign_\phi(\mathcal{D}; \mathcal{A}) = \{ \bigcup_{n \in T} \overline{B}_\varepsilon \mid \varepsilon \in a\text{-}suc\text{-}runs_{\phi,\mathcal{D};\mathcal{A}} \}$$

The input tree $\mathcal{A};\mathcal{D}$ for our automata

For input τ-structure $\mathcal{A}$ and input tree decomposition $\mathcal{D} = (\mathcal{T}, (\overline{a}_{n\in T}))$: colored tree $\mathcal{A};\mathcal{D} = (\mathcal{T}, c)$ has

1. *underlying tree* same as in the tree-decomposition
2. *color* at n corresponds to the information about
   - the bags at n and parent n' : $|a_n|$, $a_n \cap a_{n'}$
   - and substructure of $\mathcal{A}$ induced at n: $R^{\mathcal{A}}|_{a_n}$

   in terms of positions of occurrences of the corr. elements

More precisely, $c(n) = (s, P, S_1, \ldots, S_\ell) \in \Gamma_{\tau,w}$
where $\Gamma_{\tau,w} = W \times \mathcal{P}(W^2) \times \mathcal{P}(W^{r_1}) \times \cdots \times \mathcal{P}(W^{r_\ell})$,
$W = \{1, \ldots, w+1\}$.

Very similar to tree $\mathcal{T}^*$ of Flum, Frick, Grohe, 2002.

Example

$\mathcal{G}$ with edge relation

$$E^{\mathcal{G}} = \{\overbrace{(a,b),(b,a),(b,e),(e,b)}^{E^{\mathcal{G}}|a_{n_1}},\overbrace{(a,d),(d,a),(d,e),(e,d)}^{E^{\mathcal{G}}|a_{n_2}},\ldots\}$$

Tree $\mathcal{G};\mathcal{D}$

$$(\underline{a},b,\underline{\underline{e}}) \mapsto (3,\emptyset,\{\overbrace{(1,2),(2,1),(2,3),(3,2)}^{E^{\mathcal{G}}|a_{n_1}}\})$$

$$(\underline{a},d,\underline{\underline{e}}) \mapsto (3,\underbrace{\{(1,1),(3,3)\}}_{a^i_{n_2}=a^i_{n_1},i=1,3},\underbrace{\{\ldots\}}_{E^{\mathcal{G}}|a_{n_2}})$$

$\ldots$

Theorem (MSO / dec-assignment automata connection)

Main Automata Theorem

For every MSO[τ]-formula $\phi(X_1, \ldots, X_k)$ and for every integer w, *there exists a w-decomposition (k-assignment) automaton w-dec-Assign$_\phi$, that running over tree $\mathcal{A}; \mathcal{D}$* where $\mathcal{A}$ is a τ-structure with treewidth w and $\mathcal{D} = (\mathcal{T}, (\overline{a}_n)_{n \in T})$ is an ordered tree-decomposition of $\mathcal{A}$ with width w *computes the satisfying assignments of ϕ over $\mathcal{A}$.* i.e.

$$w\text{-dec-Assign}_\phi(\mathcal{A}; \mathcal{D}) = \phi(\mathcal{A})$$

Proof Sketch

- Construction of automata for atomic formulas $In(x, X)$ and $R(x_1, \ldots, x_\ell)$:
  - Number of states for $In(x, X)$: 3
  - Number of states for $E(x, y)$: $2^{w+2} + 3$
    $q_0, q_1^I, q_2^I, q_a, q_f$ for $I \subseteq W = \{1, \ldots, w+1\}$

    example:

$$\boldsymbol{a}, b, e \to q_a, (\{1\}, \emptyset)$$

$$q_2^{\{1,3\}}, (\emptyset, \{2\}) \leftarrow a, \boldsymbol{d}, e \qquad b, c \to q_0, (\emptyset, \emptyset)$$

- The construction of automata for not atomic formulas is analogous to the usual tree-automata inductive definitions

Make sure that produce proper encodings (elements of $\mathcal{A}$ are selected via their top-most occurrences).

Counting version of our automata

The run ρ of w-*dec-Assign*$_\phi^{\text{count}}$ over $\mathcal{T}_{\mathcal{I};\mathcal{A}}$ is a mapping

$$T \to Q \times \mathcal{P}(W) \times \{1, \ldots, |A|\}$$

Three variants of MSO evaluation problems that require evaluating the size of sets involved:

1. "$\psi^{\exists,k} = \exists^k X \phi(X)$" which is true if there exists a set X of size k satisfying MSO formula $\phi(X)$.
2. "$\text{count-}\phi^{\min/\max}(X)$" whose (unique) satisfying assignment is the size of the sets satisfying $\phi^{\min/\max}(X)$
3. "$\psi(X) = \phi^{\min/\max}(X) = \min/\max X : \phi(X)$" whose satisfying assignments are those satisfying assignments of the MSO formula ϕ having minimum/maximum size,

From MSO to datalog

the idea of the reduction

MSO-formula ϕ evaluation	$- - - - - - - \rightarrow$ $[w\text{-}dec\text{-}Assign_\phi]$	datalog query Q_ϕ evaluation

- Consider automaton w-*dec-Assign*$_\phi$
- Given $\mathcal{A}$ and $\mathcal{D}$, program $\Pi_{\tau,w}$ produces $\mathcal{A};\mathcal{D}$
- Program Π_ϕ compute assignments encoded in runs of w-*dec-Assign*$_\phi$ over $\mathcal{A};\mathcal{D}$ performing component-wise unions at each node n:

$$\overline{B}_{\varepsilon,T_n} = \overline{B}_{\varepsilon,n} \cup \overline{B}_{\varepsilon,T_{n_1}} \cup \overline{B}_{\varepsilon,T_{n_2}}$$

Then it selects the ones corresponding to successful runs

- The final datalog solution is a result of **rewriting optimization method for datalog** (known as "magic sets" algorithm); thus we obtain **optimized programs** excluding the computation of non satisfying.
- The resulting program correspond an efficient automata evaluation algorithm that coincides with Flum, Frick, Grohe algorithm.
- Moreover we define automata construction and minimization in datalog, we thus obtain a fully database theoretic i.e. **given ϕ, $\mathcal{A}$, $\mathcal{D}$ there exits a datalog query computing $\phi(\mathcal{A})$ thats does not requires the automaton since it is able to construct it.**

The optimized program Π_ϕ^S

The optimized datalog program Π_ϕ^S

I. (Optimum) filter program defining predicates *suc-q*
successful states of w-*dec-Assign*$_\phi$ are computed

II. r_{Δ_0} : *assign-*$q(x,\overline{x}) \leftarrow$ *suc-*$q(x)$, *color-*$\gamma(x)$, *leaf*(x), *decomp-*$\overline{I}(x,\overline{x})$

r_Δ : *assign-*$q(x,\overline{x}) \leftarrow$ *suc-*$q(x)$, *color-*$\gamma(x)$, *succ*$_0(x,x_1)$, *assign-*$q_1(x_1,\overline{y})$, *succ*$_1(x,x_2)$, *assign-*$q_2(x_2,\overline{z})$ *decomp-*$\overline{I}(x,\overline{v})$, *cw-union*$(\overline{x},\overline{v},\overline{y},\overline{z})$

r_F : ϕ-*assign*$(\overline{e}) \leftarrow$ *assign-*$q(x,\overline{x})$

restrictions of satisfying assignments are computed

Datalog results

Theorem

Let $\tau = \{R_1, \ldots, R_\ell\}$; for every $\mathrm{MSO}[\tau]$ formula $\phi(X_1, \ldots . X_k)$ and integer w, we can construct a datalog$^\neg$ query $Q^w_{\phi,k} = (\Pi^w_{\phi,k}, \textit{assign})$ of arity $k+1$ s. t. for every τ-structure $\mathcal{A}$ with special tree-decomposition $\mathcal{D}$ of width w,

$$(B_1, \ldots, B_k) \in \phi(\mathcal{A}) \text{ iff } (B_1, \ldots, B_k) \in Q^w_\phi(\mathcal{D}^U_{\mathcal{A},\mathcal{I}})$$

Complexity at most $(\ell + 2 + 2^{k\cdot(w+1)+1} \cdot s^2 + |\phi(\mathcal{A})|) \cdot |T| + |\phi(\mathcal{A})|$.

Theorem

The class of k-ary MSO-definable queries over relational structures of bounded treewidth is $(k+1)$-datalog definable.

"MSO evaluation problems involving counting": evaluation via datalog

Three variants that require evaluating the size of sets involved:

1. "$\psi^{\exists,k} = \exists^k X \phi(X)$" : **Optimized monadic datalog program**
   Complexity $(\ell + 2 + 2^{w+1} \cdot s^2 + 1) \cdot |T|$.
   (one traversal of the tree-decomposition)
2. "$\text{count-}\phi^{\min/\max}(X)$": **Monadic datalog program**
3. "$\psi(X) = \phi^{\min/\max}(X) = \min/\max X\phi(X)$" : **Optimized datalog program of arity 2**

Extend datalog with predicates corresponding to arithmetic operations over integers.

The vertex cover automata

Table: Deterministic automaton for formula $\phi(Z) \equiv$ "Z is a vertex cover".

$q_1, q_2, (\gamma, I) \mapsto q,\ I, S, L \subseteq W;\ F = \{Q^{\emptyset}\}$

Transition	**Conditions satisfied at current node n**
$\gamma, \emptyset / L \mapsto Q^I$	$NotEdge_\gamma$ & $AdjInPar_\gamma = I /$ $L \neq \emptyset$ & $set(L)$ & $NotEdge_{\gamma \setminus L}$ & $AdjInPar_{\gamma \setminus L} = I$
$Q^J, Q^S, \gamma, \emptyset \mapsto Q^I$	$NotEdge_\gamma$ & $(J \cup S) \cap V_\gamma = \emptyset$ & $SameInPar(J \cup S) \cup AdjInPar_\gamma = I$
$Q^J, Q^S, \gamma, L \mapsto Q^I$	$L \neq \emptyset$ & $set(L)$ & $NotEdge_{\gamma \setminus L}$ & $(J \cup S) \cap (V_\gamma \setminus L) = \emptyset$ & $SameInPar(J \cup S) \cup AdjInPar_{\gamma \setminus L} = I$
$q_1, q_2, \gamma, I \mapsto Q_f$	all cases that were not considered above