Space and Circuit Complexity of Monadic Second-Order Definable Problems on Tree-Decomposable Structures

Michael Elberfeld

University of Lübeck

Finite and Algorithmic Model Theory 2012
Les Houches, May 14, 2012

1. Computational Complexity of MSO-Definable Problems
2. ... on Tree-Width-Bounded Structures
3. ... on Tree-Depth-Bounded Structures

Many Problems are MSO-Definable

Definition (MSO-Definable Problem)

Let $\varphi(X)$ be an MSO-formula.

Given Graph G

Decide $G \models \exists X\, \varphi(X)$

Count $S \subseteq V$ with $G \models \varphi(S)$

Many Problems are MSO-Definable

Definition (MSO-Definable Problem)

Let $\varphi(X)$ be an MSO-formula.

Given	Graph G
Decide	$G \models \exists X\, \varphi(X)$
Count	$S \subseteq V$ with $G \models \varphi(S)$

Example (Decide 3-Colorable Graphs)

$$\varphi = \exists R, G, B \;\; \forall v\, [R(v) \vee G(v) \vee B(v)] \;\; \forall v, w\, \big[E(v,w) \to \neg[R(v) \wedge R(w)] \wedge \neg[G(v) \wedge G(w)] \wedge \neg[B(v) \wedge B(w)]\big]$$

We have  $\models \varphi$, but  $\not\models \varphi$.

Many Problems are MSO-Definable

Definition (MSO-Definable Problem)

Let $\varphi(X)$ be an MSO-formula.

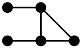
Given Graph G

Decide $G \models \exists X\, \varphi(X)$

Count $S \subseteq V$ with $G \models \varphi(S)$

Example (Count Dominating Sets)

$$\varphi(X) = \forall v \left[X(v) \vee \exists w[E(v,w) \wedge X(w)]\right]$$

There are 15 vertex sets S with  $\models \varphi(S)$.

How Hard are MSO-Definable Problems?

Situation on General Structures

Decision can be NP-hard (take 3-COLORABILITY)

Counting can be #P-hard (take #PERFECT-MATCHING)

How Hard are MSO-Definable Problems?

Situation on General Structures

Decision can be NP-hard (take 3-COLORABILITY)

Counting can be #P-hard (take #PERFECT-MATCHING)

Situation on Tree-Decomposable Structures

Algorithmics Solvable in

- sequential linear time [Courcelle, 1990]
- parallel logarithmic time [Bodlaender and Hagerup, 1998]

Complexity Problems lie in P, but

how deep inside P can we place them?

How Hard are MSO-Definable Problems?

Situation on General Structures

Decision can be NP-hard (take 3-COLORABILITY)

Counting can be #P-hard (take #PERFECT-MATCHING)

Situation on Tree-Decomposable Structures

Algorithmics Solvable in

- sequential linear time [Courcelle, 1990]
- parallel logarithmic time [Bodlaender and Hagerup, 1998]

Complexity Problems lie in P, but

how deep inside P can we place them?

Why Studying?

Unify Problems Applications to a wide range of problems.

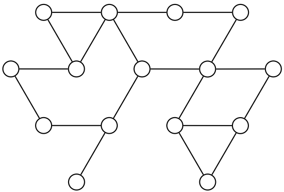
Unify Techniques Drives development of general techniques.

❷ . . . on Tree-Width-Bounded Structures

What are Tree-Width-Bounded Graphs?

Definition (Tree Decomposition)

Graph

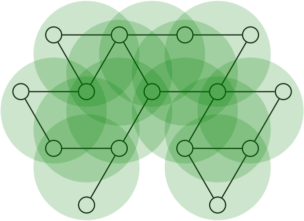


Decomposition

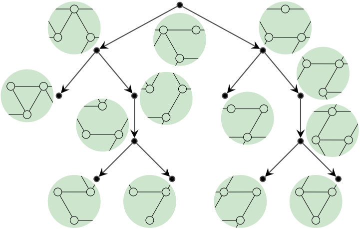
What are Tree-Width-Bounded Graphs?

Definition (Tree Decomposition)

Graph



Decomposition

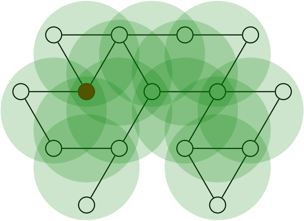


- Bags cover all vertices and edges

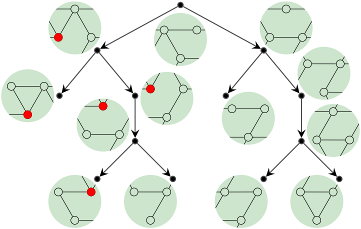
What are Tree-Width-Bounded Graphs?

Definition (Tree Decomposition)

Graph



Decomposition

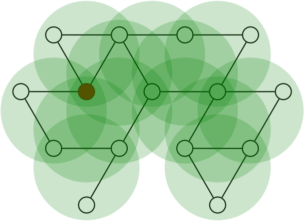


- Bags cover all vertices and edges
- For every vertex, its bags are connected

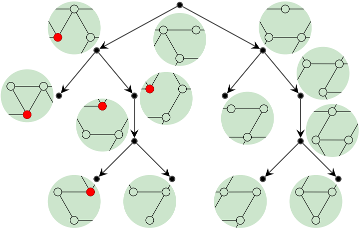
What are Tree-Width-Bounded Graphs?

Definition (Tree Decomposition)

Graph



Decomposition



- Bags cover all vertices and edges
- For every vertex, its bags are connected

Definition (Bounded Tree Width)

A class *C* of graphs has bounded tree width if its graphs have tree decompositions with

- size-bounded bags.

Complexity of MSO-Problems

Definition (MSO-Problem on Tree-Width-Bounded Graphs)

Let $\varphi(X)$ be an MSO-formula, C class of bounded tree width.

- **Given** Graph $G \in C$
- **Decide** $G \models \exists X \, \varphi(X)$
- **Count** $S \subseteq V$ with $G \models \varphi(S)$

Complexity of MSO-Problems

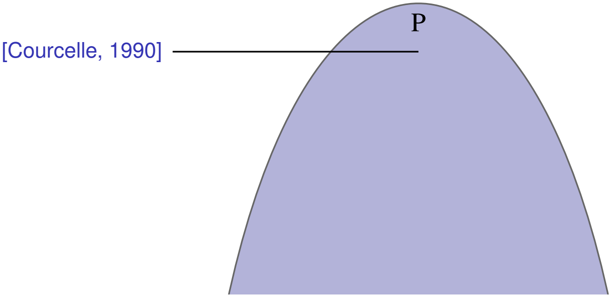
Definition (MSO-Problem on Tree-Width-Bounded Graphs)

Let $\varphi(X)$ be an MSO-formula, C class of bounded tree width.

Given Graph $G \in C$

Decide $G \models \exists X \, \varphi(X)$

Count $S \subseteq V$ with $G \models \varphi(S)$



Complexity of MSO-Problems

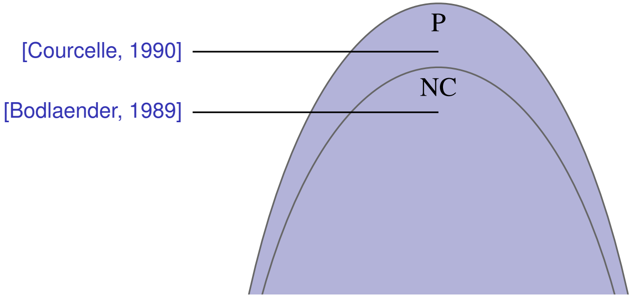
Definition (MSO-Problem on Tree-Width-Bounded Graphs)

Let $\varphi(X)$ be an MSO-formula, C class of bounded tree width.

Given Graph $G \in C$

Decide $G \models \exists X \, \varphi(X)$

Count $S \subseteq V$ with $G \models \varphi(S)$

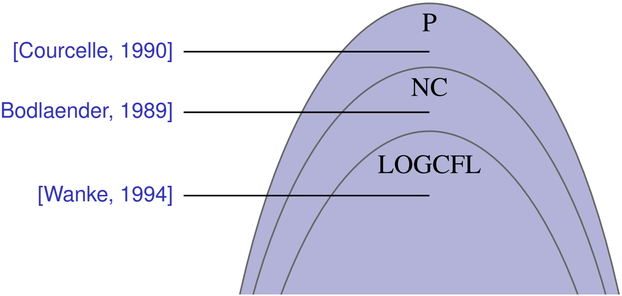


Complexity of MSO-Problems

Definition (MSO-Problem on Tree-Width-Bounded Graphs)

Let $\varphi(X)$ be an MSO-formula, C class of bounded tree width.

Given	Graph $G \in C$
Decide	$G \models \exists X \, \varphi(X)$
Count	$S \subseteq V$ with $G \models \varphi(S)$



Complexity of MSO-Problems

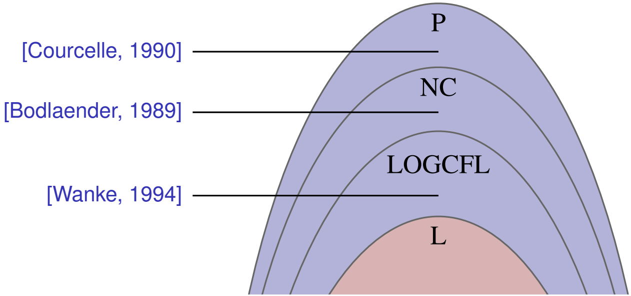
Definition (MSO-Problem on Tree-Width-Bounded Graphs)

Let $\varphi(X)$ be an MSO-formula, C class of bounded tree width.

Given Graph $G \in C$

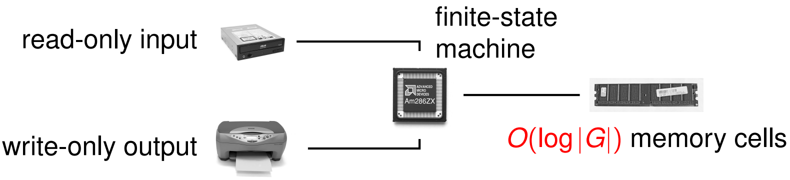
Decide $G \models \exists X \, \varphi(X)$

Count $S \subseteq V$ with $G \models \varphi(S)$



The World of Logspace

Definition (Logarithmic-Space-Bounded Machine)



Fact (Algorithmic Toolbox via Recomputations)

Logspace is closed under

- *composition,* *[Stockmeyer and Meyer, 1973]*
- *subprocedure calls* *[Ladner and Lynch, 1976]*

Logspace Upper Bound

Theorem (E., Jakoby and Tantau, FOCS 2010)

For any MSO*-formula* $\varphi(X)$ *and graphs of bounded tree width,*

Deciding $G \models \exists X\, \varphi(X)$ *lies in* L

Counting $S \subseteq V$ *with* $G \models \varphi(S)$ *lies in* L

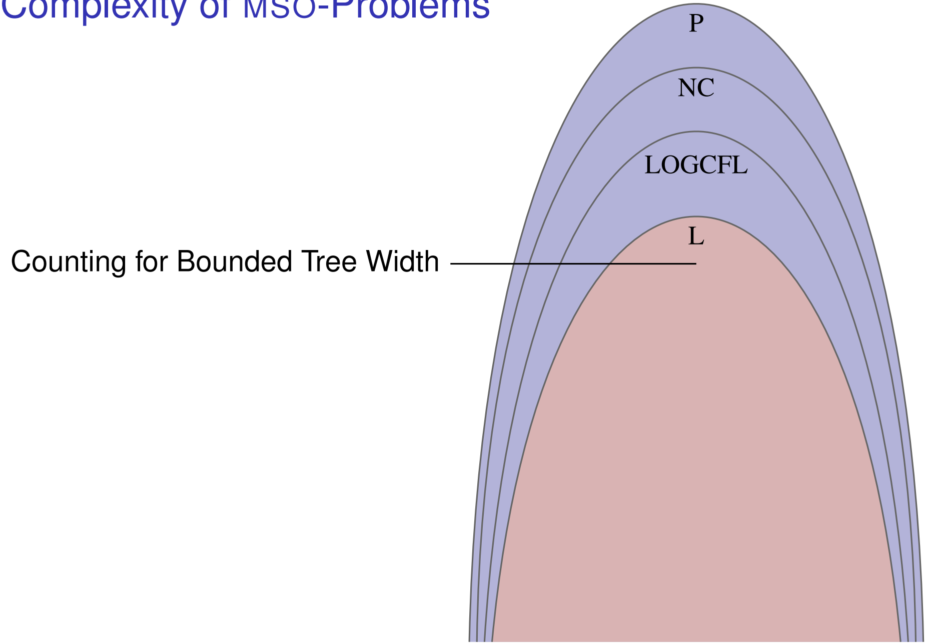
Applications to Hard Problems

- For tree-width-bounded graphs, REACHABILITY and PERFECT-MATCHING $\in \mathrm{L}$
- EVEN-CYCLE $= \{$ simple $G \mid G$ has even-length cycle $\} \in \mathrm{L}$

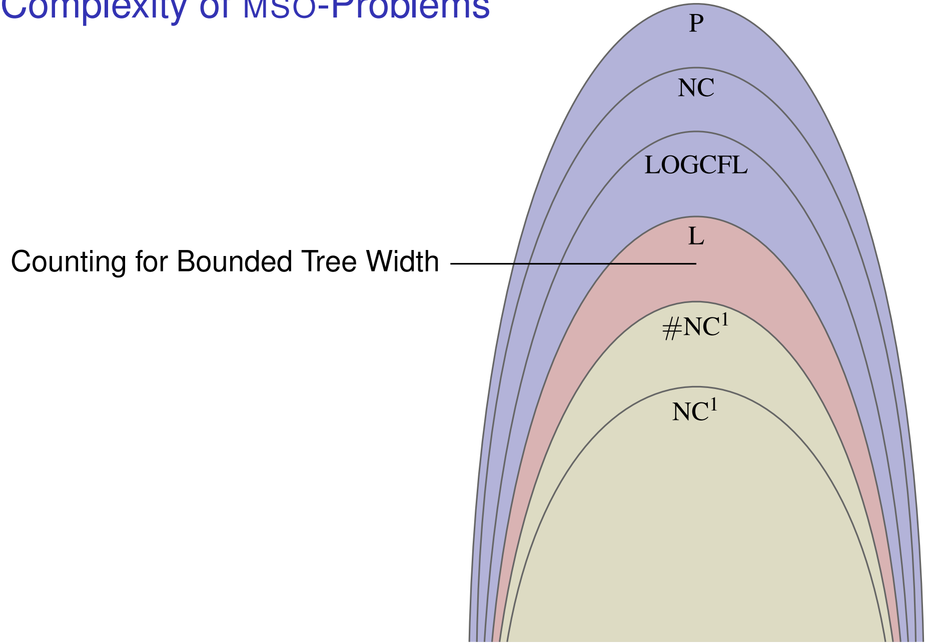
Proof Plan.

1. Compute tree decomposition of bounded width.
2. Turn φ into an equivalent automaton, and evaluate it. □

Complexity of MSO-Problems



Complexity of MSO-Problems



The World of Logarithmic-Depth Circuits

Definition (Logarithmic-Depth Bounded Fan-In Circuits)

NC^1 Boolean $\{\vee, \wedge\}$-gates over $x_1, \ldots x_n, \neg x_1, \ldots, \neg x_n$

$\#\mathrm{NC}^1$ Arithmetic $\{+, \cdot\}$-gates over $x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n$

The World of Logarithmic-Depth Circuits

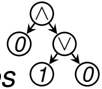
Definition (Logarithmic-Depth Bounded Fan-In Circuits)

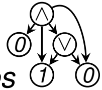
NC^1 Boolean $\{\vee, \wedge\}$-gates over $x_1, \ldots x_n, \neg x_1, \ldots, \neg x_n$

$\#\mathrm{NC}^1$ Arithmetic $\{+, \cdot\}$-gates over $x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n$

Fact (Main Barrier When Moving From L to NC^1)

BOOLEAN-SENTENCE-EVALUATION *is*

- L*-complete for pointer structures* 

.
- NC^1*-complete for terms* $(0 \wedge (1 \vee 0))$ *or closures* 

.

Logarithmic-Depth Upper Bound

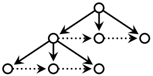
Theorem (E., Jakoby and Tantau, STACS 2012)

For any MSO*-formula* $\varphi(X)$ *and graphs given with*
width-bounded tree decompositions and their transitive closure

Deciding $G \models \exists X\, \varphi(X)$ *lies in* NC^1

Counting $S \subseteq V$ *with* $G \models \varphi(S)$ *lies in* $\#\mathrm{NC}^1$

Applications to Complete Problems

- BOOLEAN-SENTENCE-EVALUATION $\in \mathrm{NC}^1$,
  ARITHMETIC-SENTENCE-EVALUATION $\in \#\mathrm{NC}^1$
- Simulate Automata for  Trees $\in \mathrm{NC}^1, \#\mathrm{NC}^1$
- Simulate Visible Pushdown Automata in $\in \mathrm{NC}^1, \#\mathrm{NC}^1$

Logarithmic-Depth Upper Bound

Theorem (E., Jakoby and Tantau, STACS 2012)

For any MSO*-formula* $\varphi(X)$ *and graphs given with*
width-bounded tree decompositions and their transitive closure

Deciding $G \models \exists X\, \varphi(X)$ *lies in* NC^1

Counting $S \subseteq V$ *with* $G \models \varphi(S)$ *lies in* $\#\mathrm{NC}^1$

Buss's Standard Proof Plan

Simultaneously,

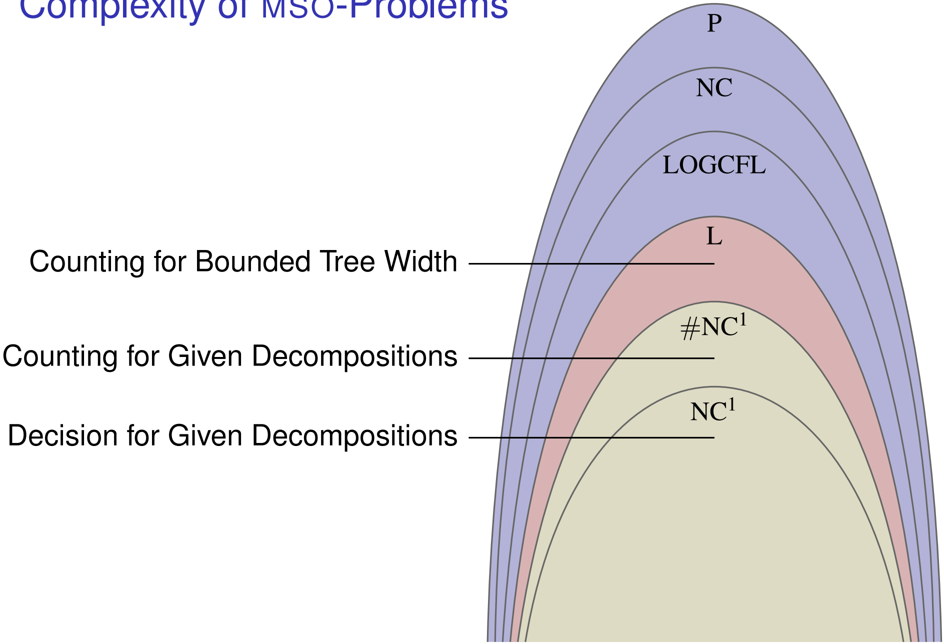
- divide input into parts of almost the same size, and
- combine intermediate results algebraically.

Decomposition-Based Proof Plan.

Successively,

1. Make decomposition balanced and binary.
2. Turn φ into an equivalent automaton, and evaluate it. □

Complexity of MSO-Problems



❸ . . . on Tree-Depth-Bounded Structures

What are Tree-Depth-Bounded Graphs?

Definition (Bounded Tree Depth)

A class *C* of graphs has bounded tree depth if its structures have tree decompositions with

- bounded-size bags, and
- depth-bounded trees.

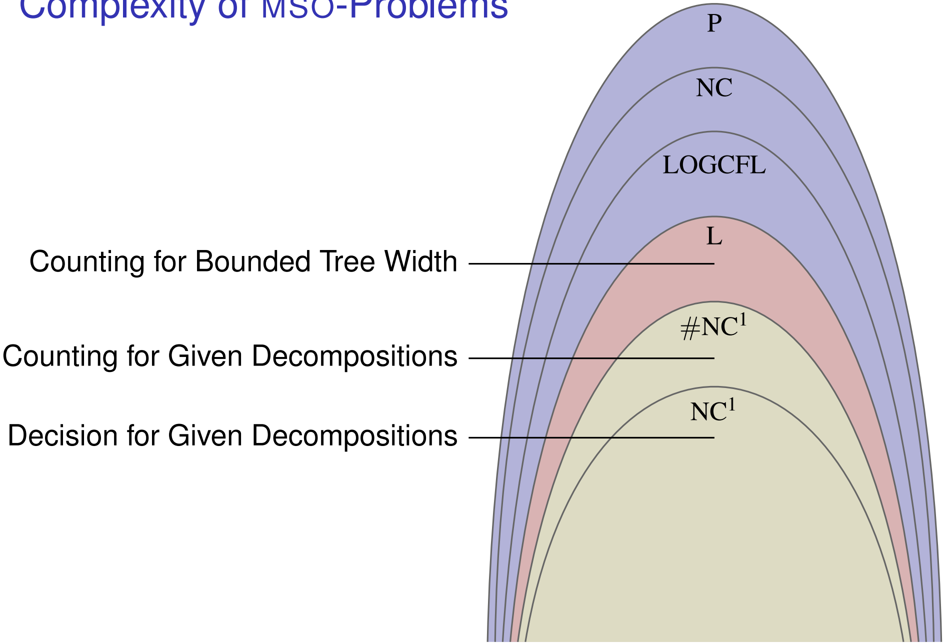
Example (Bounded Tree Depth)

- The class of all  .

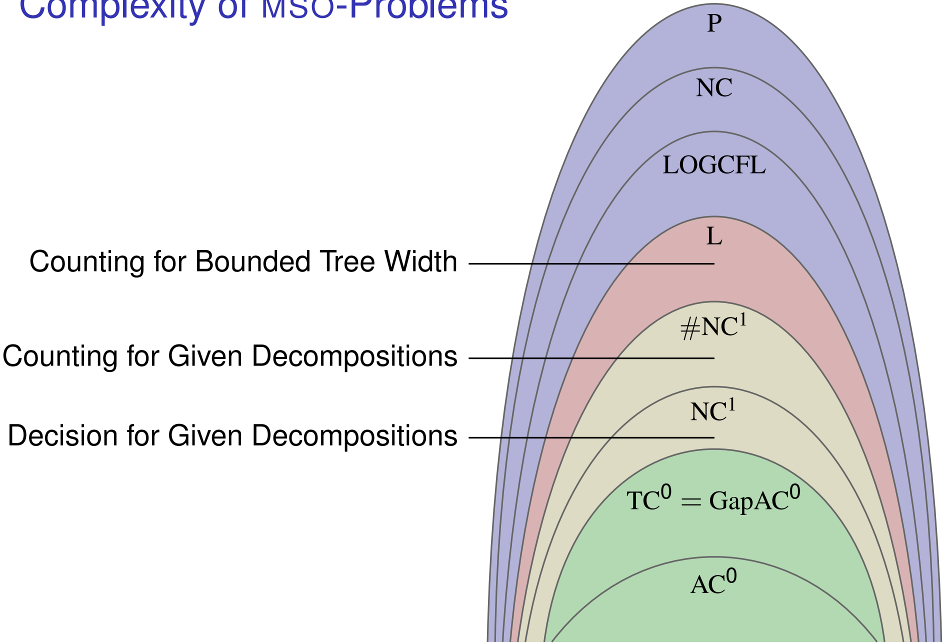
Example (Unbounded Tree Depth)

- The class of all  .
- The class of all 
  (characterizes unbounded tree depth).

Complexity of MSO-Problems



Complexity of MSO-Problems



The World of Constant-Depth Circuits

Definition (Constant-Depth Unbounded Fan-In Circuits)

AC^0 Boolean $\{\vee, \wedge\}$-gates over $x_1, \ldots x_n, \neg x_1, \ldots, \neg x_n$

GapAC^0 Arithmetic $\{+, -, \cdot\}$-gates over $x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n$

Fact (Back and Forth Between Numbers and Strings)

Evaluating GapAC^0*-circuits can be done in* TC^0*, and vice versa.*

[Hesse et al., 2002]

Constant-Depth Upper Bounds

Theorem (E., Jakoby and Tantau, STACS 2012)

For any MSO*-formula* $\varphi(X)$ *and graphs of bounded tree depth,*

Deciding $G \models \exists X\, \varphi(X)$ *lies in* AC^0

Counting $S \subseteq V$ *with* $G \models \varphi(S)$ *lies in* GapAC^0

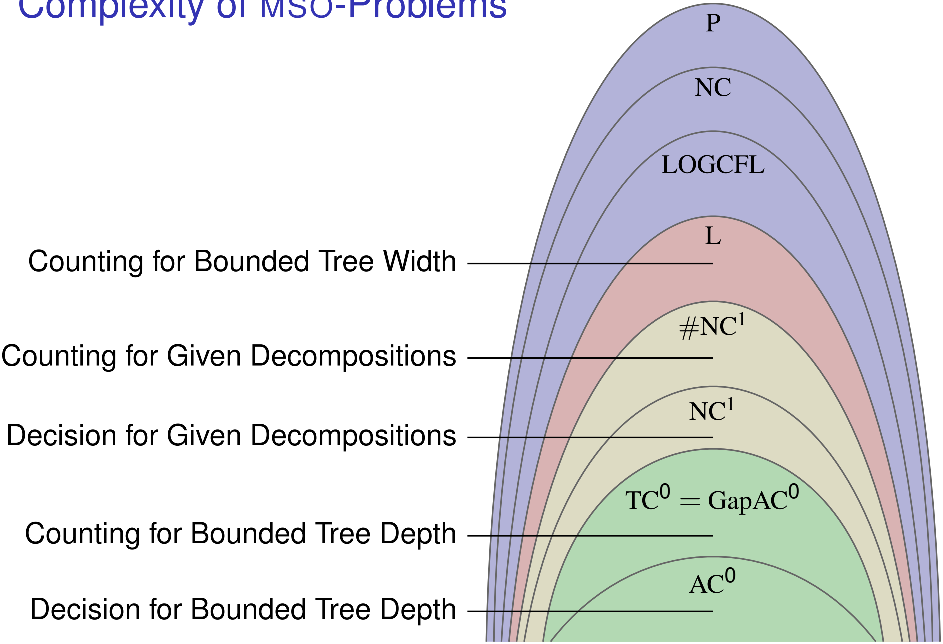
Applications to Hard Problems

- For tree-depth-bounded graphs,
  $\#$PERFECT-MATCHINGS $\in \mathrm{GapAC}^0$
- UNARY-KNAPSACK, ... $\in \mathrm{TC}^0$ via a similar theorem.

Proof Plan.

1. Compute decomposition of bounded width and depth.
2. Turn φ into an automaton for unbounded-degree trees, and evaluate automaton using arithmetic circuits. □

Complexity of MSO-Problems



Outlook

Larger Graph Classes

Consider more general graphs. Move from ...

- tree width to clique width
- tree depth to ???

Other Logics

Consider more expressive logics. Move from ...

- MSO to counting MSO

Refine Resource Bounds

Work on structures directly, not their encoding. Move from ...

- $\mathrm{AC}^0 = \mathrm{FO}$ to first-order logic [E., Grohe, Tantau, LICS 2012]
- $\mathrm{NC}^1 = \mathrm{FO}[\mathrm{REG}]$ to ???
- $\mathrm{L} = \mathrm{FO}(\mathrm{DTC})$ to ???

References I

Bodlaender, H. L. (1989).
NC-algorithms for graphs with small treewidth.
In *Proceedings of the 14th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 1988)*, volume 344 of *Lecture Notes in Computer Science*, pages 1–10. Springer.

Bodlaender, H. L. and Hagerup, T. (1998).
Parallel algorithms with optimal speedup for bounded treewidth.
SIAM Journal on Computing, 27(6):1725–1746.

Courcelle, B. (1990).
Graph rewriting: An algebraic and logic approach.
In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 193–242. Elsevier and MIT Press.

References II

Hesse, W., Allender, E., and Mix Barrington, D. A. (2002).
Uniform constant-depth threshold circuits for division and iterated multiplication.
Journal of Computer and System Sciences, 65(4):695–716.

Ladner, R. E. and Lynch, N. A. (1976).
Relativization of questions about log space computability.
Theory of Computing Systems, 10:19–32.

Stockmeyer, L. J. and Meyer, A. R. (1973).
Word problems requiring exponential time (preliminary report).
In *Proceedings of the 5th annual ACM symposium on Theory of computing (STOC 1973)*, pages 1–9. ACM.

Wanke, E. (1994).
Bounded tree-width and LOGCFL.
Journal of Algorithms, 16(3):470–491.

Copyright Notes for Images