

Context-Free Graph Properties via Definable Decompositions

Michael Elberfeld (paper author)

Kord Eickmeyer (gives talk)

RWTH Aachen University

CSL 2016

Marseille, 30 August, 2016

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

MSO-logic is used outside model theory.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Monadic second-order (MSO) logic

Syntax φ quantifies sets $\exists X \subseteq U$ besides $\exists x \in U$,

Semantics defines property $\{\text{finite structure } \mathcal{A} \mid \mathcal{A} \models \varphi\}$.

Beyond Monadic
Second-Order Logic

Strings of even length are MSO-definable

Language $L = \{a^n \mid n \text{ is even}\}$ is MSO-definable since

Property $\text{STRUCTURE}(L) = \{\circ \rightarrow \circ, \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ, \dots\}$ is
MSO-definable via guess-and-check **coloring**.

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Language-Theoretic Fact

Regular languages are MSO-definable, and vice versa.

[Büchi, 1960, Elgot, 1961, Trakhtenbrot, 1961]

MSO-logic is used outside model theory.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Monadic second-order (MSO) logic

Syntax φ quantifies sets $\exists X \subseteq U$ besides $\exists x \in U$,

Semantics defines property $\{\text{finite structure } \mathcal{A} \mid \mathcal{A} \models \varphi\}$.

Beyond Monadic
Second-Order Logic

Strings of even length are MSO-definable

Language $L = \{a^n \mid n \text{ is even}\}$ is MSO-definable since

Property $\text{STRUCTURE}(L) = \{ \bullet \rightarrow \bullet, \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet, \dots \}$ is
MSO-definable via guess-and-check **coloring**.

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Language-Theoretic Fact

Regular languages are MSO-definable, and vice versa.

[Büchi, 1960, Elgot, 1961, Trakhtenbrot, 1961]

MSO-logic is used outside model theory.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Monadic second-order (MSO) logic

Syntax φ quantifies sets $\exists X \subseteq U$ besides $\exists x \in U$,

Semantics defines property $\{\text{finite structure } \mathcal{A} \mid \mathcal{A} \models \varphi\}$.

Beyond Monadic
Second-Order Logic

3-colorable graphs are MSO-definable

We have  $\models \varphi$, but  $\not\models \varphi$ with MSO-formula

$$\varphi = \exists R, G, B \subseteq U \\ \forall v \in U [v \in R \vee v \in G \vee v \in B] \wedge \\ \forall (v, w) \in E \neg [v, w \in R \vee v, w \in G \vee v, w \in B]$$

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Algorithmic Fact

Each MSO-definable property is solvable in *linear time*
on structures with *bounded tree width*. [Courcelle, 1990]

Second-order logic is too general.

Second-order (SO) logic

Syntax φ uses higher-ary quantifiers $\exists X \subseteq U \times U$,

Semantics defines property $\{\text{finite structure } \mathcal{A} \mid \mathcal{A} \models \varphi\}$.

Complexity Fact

*The SO-definable properties of strings are the problems of the **polynomial hierarchy** (PH).* [Stockmeyer, 1976]

SO's expressivity and complexity

Since $P \subseteq NP \subseteq PH \subseteq PSPACE$,

- SO-definable properties are far beyond regular and context-free, and
- modest polynomial runtimes are out of reach.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

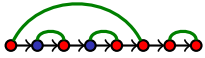
A matching relation contains nested tuples.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

String structures with matchings

Structure $(\mathcal{S}, \mathcal{M}) =$  representing
 $s = 10101111$ and well-formed $m = (() ()) ()$.

The logic “ \exists Match MSO” for string structures

Syntax $\psi = \exists^{\text{matching}} M [\varphi]$ with MSO-formula φ .

Semantics String structure $\mathcal{S} \models \psi$ if
 $(\mathcal{S}, \mathcal{M}) \models \varphi$ for a matching \mathcal{M} of \mathcal{S} .

Language-Theoretic Fact

*The “ \exists Match MSO”-definable properties of strings are the **context-free** languages.*

[Lautemann, Schwentick, and Thérien, 1995]

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Paper generalizes from strings to graphs.

Research question

“a more general study of these logics will prove worthwhile in the context of general finite structures, instead of strings.”

[Lautemann, Schwentick, and Thérien, 1995]

Paper's contribution

A logic for **structures with bounded tree width** along with understanding the following aspects:

- languages** Its equivalence to context-free languages.
- model theory** Its expressive power.
- algorithms** Its satisfiability and evaluation problems.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

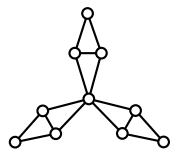
Aspect I:
Formal languages

Aspect II:
Model Theory

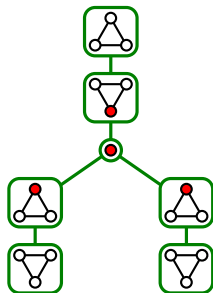
Aspect III:
Algorithm Design

Conclusion

We use decompositions of bounded width.



graph \mathcal{G}



tree decomposition \mathcal{D}

Properties of our tree decompositions

Connected For **every vertex**, its bags are connected.

Covering **Bags** cover all edges

Strictness Subtrees cover ever “smaller” subgraphs.

Normal Subtrees cover connected subgraphs.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

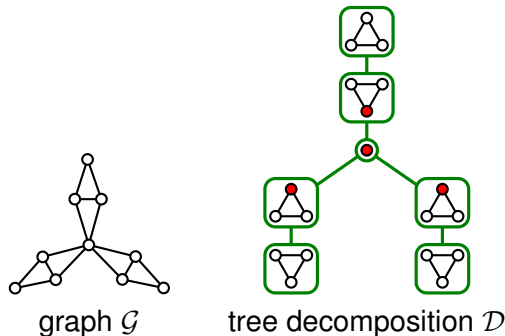
Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

We use decompositions of bounded width.



Width of tree decompositions

$\text{width}(\mathcal{D}) = \text{maximum bag size} - 1.$

Graphs with width-bounded decompositions

- Paths and trees have width-1 decompositions,
- Series-parallel graphs have width-2 decompositions.

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Different tree decompositions are possible.

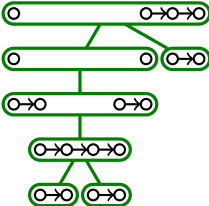
Flexibility of tree decompositions

- Many width-optimal decompositions exist in general.
- More apparent with a slightly higher width.

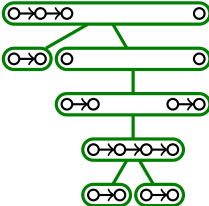
Width-3 decompositions of a string structure

○→○→○→○→○→○→○→○→○

decomposition \mathcal{D}_1



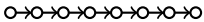
decomposition \mathcal{D}_2

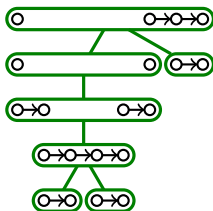


...

Our logic adds quantifying decompositions.

Structures expanded by decompositions

Combining string structure \mathcal{S} and decomposition \mathcal{D} into a structure like $(\mathcal{S}, \mathcal{D}) =$ 



Decomposition MSO (DMSO) for structures

Syntax $\psi = \exists^{\text{width} \leq w} D[\varphi]$ with MSO φ and $w \in \mathbb{N}$.

Semantics $\mathcal{A} \models \psi$ if $(\mathcal{A}, \mathcal{D}) \models \varphi$ for a width- w tree decomposition \mathcal{D} of \mathcal{A} .

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

DMSO on strings captures context-freeness.

Context-Free
Graph Properties
via Definable
Decompositions

Theorem


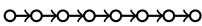
The DMSO-definable properties of strings are the *context-free* languages.

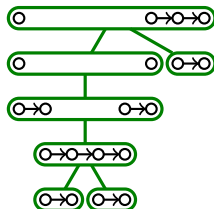
Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Proof relates matchings and decompositions.

- Define CFLs in terms of “ \exists Match MSO”-logic.

- From  $\models \varphi$ to  $\models \varphi'$.



- From $(S, \mathcal{D}) \models \varphi'$ to $(S, \mathcal{M}) \models \varphi$.

□

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Higher width means higher expressivity.

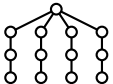
Recall: Syntax of DMSO-logic

DMSO-formulas are $\psi = \exists^{\text{width} \leq w} D[\varphi]$ with width $w \in \mathbb{N}$

Theorem

For every $w \geq 3$ and $w' \leq w - 2$, formulas with width w are more expressive than formulas with width w' .

Proving subset and not-equal relation.

- “ \subseteq ”
- Width- w cover width- w' decompositions.
 - Restricting to width- w' is MSO-definable.
- “ \neq ”
- Width $w = 4$ allows to define subdivided stars with 4 paths of equal length like , but
 - is not possible with $w = 2$ (via Ehrenfeucht-Fraïssé games). □

Satisfiability for DMSO-formulas is decidable.

Context-Free
Graph Properties
via Definable
Decompositions

Theorem

Testing whether a given DMSO-formula
 $\psi = \exists^{\text{width} \leq w} D [\varphi]$ *has an* \mathcal{A} *with* $\mathcal{A} \models \psi$ *is decidable.*

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Proof is based on Seese's Theorem.

- To test if there is \mathcal{A} with $\mathcal{A} \models \psi$, we test if there is \mathcal{A} and width- w \mathcal{D} with $(\mathcal{A}, \mathcal{D}) \models \varphi$.
- Expansion $(\mathcal{A}, \mathcal{D})$ has tree width at most $w + 2$.
- Thus, satisfiability for subformula φ is decidable [Seese, 1991].
- In turn, satisfiability for formula ψ is decidable. \square

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

Conclusion

Model Checking DMSO-formulas in PTIME.

Context-Free
Graph Properties
via Definable
Decompositions

Theorem

Each DMSO-definable property is decidable in polynomial time.

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Proof is based on dynamic programming.

- Let $\psi = \exists^{\text{width} \leq w} D[\varphi]$ be a formula for the property.
- We only need to look at width- w decompositions.

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO

Aspect I:
Formal languages

Aspect II:
Model Theory

Aspect III:
Algorithm Design

recursion Try all root bags and all ways of extending it.
Strictness implies exponential time.

dynamic Bottom-up evaluation of recursive calls.
Normal condition implies polynomial time. \square

Conclusion

Conclusion

Summary

- DMSO-logic is proposed to capture the notion of context-freeness for graphs.
- Basic expressibility and closure results.
- Satisfiability is decidable, model checking in PTIME.

Outlook

- Compare DMSO-logic to grammar-based notions of context-freeness for graphs?
- Model checking . . .
 - ... **fixed-parameter tractable** with parameter $|\psi|$?
 - ... in **LOGCFL** \subseteq $\text{NC}^2 \subseteq$ PTIME for each fixed ψ ?

Context-Free
Graph Properties
via Definable
Decompositions

Michael Elberfeld
(paper author)

Kord Eickmeyer
(gives talk)

Beyond Monadic
Second-Order Logic

Our Logic:
Decomposition
MSO




Aspect I:
Formal languages

Aspect II:
Model Theory




Aspect III:
Algorithm Design

Conclusion

References I

-  Büchi, J. R. (1960).
Weak second-order arithmetic and finite automata.
Math. Logic Quart., 6(1–6):66–92.
-  Courcelle, B. (1990).
Graph rewriting: An algebraic and logic approach.
In *Handbook of Theoretical Computer Science, Volume B*, pages 193–242. Elsevier and MIT Press.
-  Elgot, C. C. (1961).
Decision problems of finite automata design and
related arithmetics.
Transactions of the American Mathematical Society,
98(1):21–51.

References II

-  Seese, D. (1991).
The structure of the models of decidable monadic theories of graphs.
Annals of pure and applied logic, 53(2):169–195.
-  Stockmeyer, L. J. (1976).
The polynomial-time hierarchy.
Theoretical Computer Science, 3(1):1 – 22.
-  Trakhtenbrot, B. A. (1961).
Finite automata and logic of monadic predicates.
Doklady Akademii Nauk SSSR, 140:326–329.
In Russian.