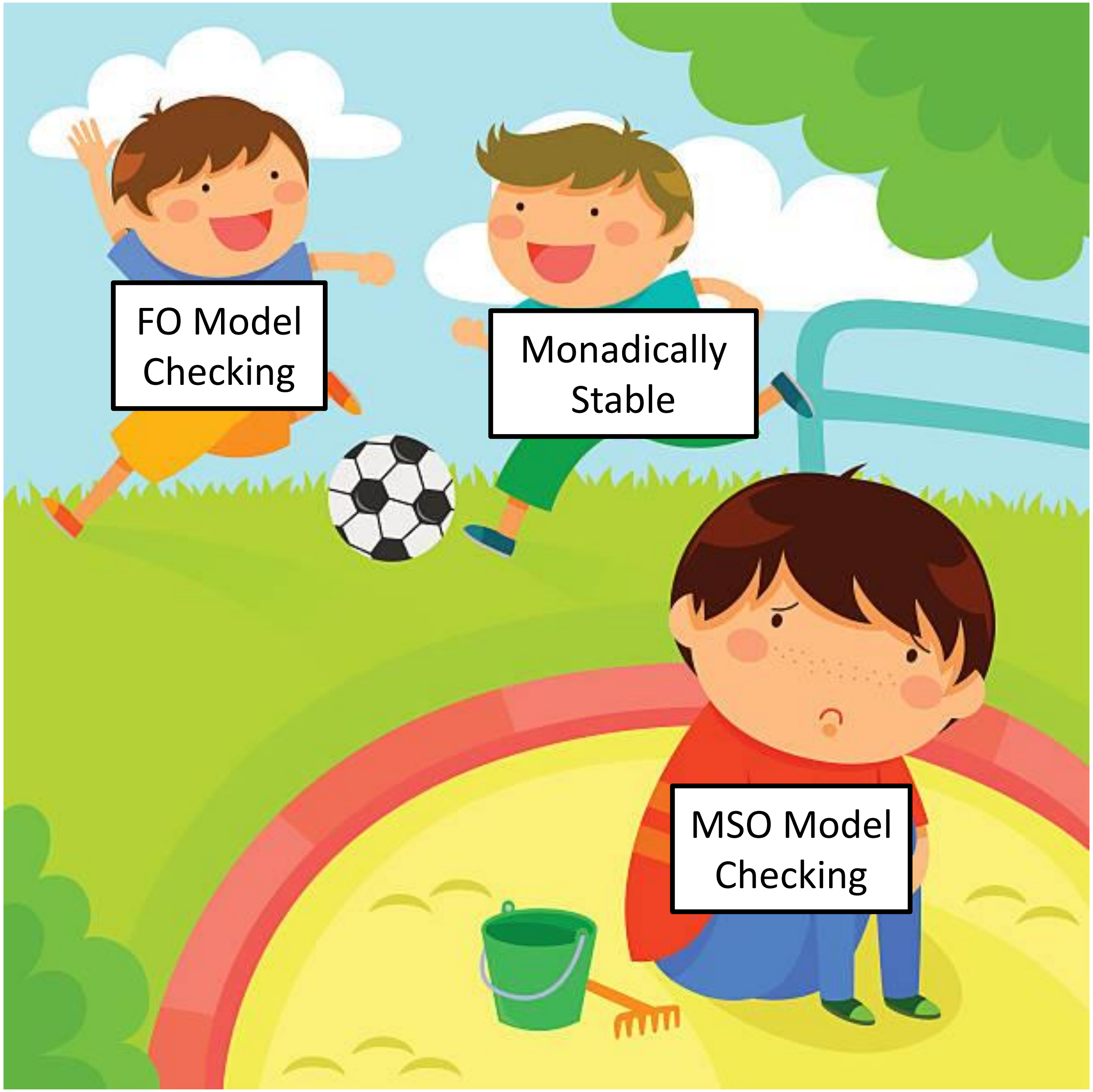
Approximate Evaluation of Quantitative Second Order Queries

Jan Dreier, **Robert Ganian**, Thekla Hamm

TU WIEN Informatics ac ALGORITHMS AND COMPLEXITY GROUP


FO Model Checking
Monadically Stable
MSO Model Checking

Motivation

- Many **NP**-hard problems are known to be fixed-parameter tractable parameterized by treewidth
  - **Minimum Vertex Cover**, **Maximum Independent Set**, **Hamiltonian Cycle**, **3-Coloring**…
  - "Standard" dynamic programming
- But what if we want to show that some new problem is fixed-parameter tractable w.r.t. treewidth?
  - Solve it via dynamic programming, or…
  - use an algorithmic meta-theorem: **Courcelle's Theorem**

Courcelle's Theorem

Given a LinCMSO_2 query **(φ, t)** and a graph **G**, we can compute an answer to **(φ, t)** on **G** in time at most $f(\boldsymbol{\varphi}, \mathrm{tw}(\mathbf{G})) \cdot |V(\mathbf{G})|$
where f is some computable function and $\mathrm{tw}(\mathbf{G})$ is the treewidth of **G**

- Why is this useful?
  - Instead of doing dynamic programming, we just need to encode problems as bounded-size "LinCMSO_2 queries"
    - Can be much, much easier

Courcelle's Theorem

Given a LinCMSO_2 query **(φ, t)** and a graph **G**, we can compute an answer to **(φ, t)** on **G** in time at most $f(\boldsymbol{\varphi}, \mathrm{tw}(\mathbf{G})) \cdot |V(\mathbf{G})|$
where f is some computable function and $\mathrm{tw}(\mathbf{G})$ is the treewidth of **G**

- **G** can be edge- and vertex-colored and weighted
- **φ** is a formula in CMSO_2 logic with some (optional) free vertex and/or edge set variables
  - Can quantify over single and set vertex/edge variables
  - Can count *modulo some fixed constant*
  - Can use usual logical connectives, query incidence etc.
- **t** is some linear function over the free variables in **φ**
- Answer: assignment which maximizes **t** or "**No**"

Courcelle's Theorem for Cliquewidth

Given a LinCMSO$_1$ query **(φ, t)** and a graph **G**, we can compute an answer to **(φ, t)** on **G** in time at most $f(\boldsymbol{\varphi}, cw(\mathbf{G})) \cdot |V(\mathbf{G})|^2$ where f is some computable function and $cw(\mathbf{G})$ is the cliquewidth of **G**

- **G** can be ~~edge- and~~ vertex-colored and weighted
- **φ** is a formula in MSO$_1$ logic with some (optional) free vertex ~~and/or edge~~ set variables
  - Can quantify over single and set vertex~~/edge~~ variables
  - Can count *modulo some fixed constant*
  - Can use usual logical connectives, query incidence etc.
- **t** is some linear function over the free variables in **φ**
- Answer: assignment which maximizes **t** or "**No**"

Our Goal

- Courcelle's Theorem(s) = best meta-theorem(s) for **exact** fixed-parameter tractability w.r.t. tw and cw
  - Not "tight": **Vertex Disjoint Paths** par. by treewidth

Our Goal

- Courcelle’s Theorem(s) = best meta-theorem(s) for **exact** fixed-parameter tractability w.r.t. tw and cw

But what about **approximate** fixed-parameter tractability w.r.t. tw and cw?

- Could capture many problems which are not **FPT**, but are **FPT**-approximable w.r.t. tw and cw (Lampis 2014)

Model Problems

- **Equitable k-Coloring / k-Connected Partition**
  - Partition graph into k equal-size parts
- **Bounded-Degree**
  - Delete a smalles
- **Capacitated Vert**
  - Cover/dominate vertices w.r.t. given capacity bounds
- **Graph Motif**
  - Find a connected subgraph in a vertex-colored graph where each color occurs precisely a given number of times

What are we approximating?

- Not only optimization problems

Model Problems

- **Equitable k-Coloring / k-Connected Partition**
  - Partition graph into k equal-size parts
- **Bounded-Degree**
  - Delete a smalles
- **Capacitated Vert**
  - Cover/dominate vertices w.r.t. given capacity bounds
- **Graph Motif**
  - Find a connected subgraph in a vertex-colored graph where each color occurs precisely a given number of times

What are we approximating?

- Not only optimization problems
- “Size constraints”

First Step: Enriching the Logic

- Goal: extend LinCMSO by allowing it to "count"
- But can't LinCMSO already count?
  - It can count *modulo some fixed constant*
    - Useful for, e.g., vertex minor detection. Not here.
- How can we let LinCMSO count?
  - Well, it already counts the optimization target **t**:

$$t(X_1 \ldots X_\ell) := a + \sum_{1 \leq i,j \leq \ell} a_{ij} w_i(X_j),$$

    - a, a_{ij} are (large input-specified) numbers, w_i is sum of weights
    - What if we add atoms that can compare these? **CMSO[$\lesseqgtr$]**

CMSO[$\lessgtr$] on Model Problems

- **Equitable k-Coloring / k-Connected Partition**
  - Partition graph into k equal-size parts
- **Bounded-Degree Vertex Deletion**
  - Delete a smallest vertex set to achieve a given max degree
- **Capacitated Vertex Cover / Dominating Set**
  - Cover/dominate vertices w.r.t. given capacity bounds
- **Graph Motif**
  - Find a connected subgraph in a vertex-colored graph where each color occurs precisely a given number of times

… But It Is Too Powerful

- **W[1]**-hard to approximate for any constant ratio α:

$$\forall Y_1 \forall Y_2 \big(|Y_1| > |Y_2| \cdot \alpha^2 \vee |Y_2| > |Y_1| \cdot \alpha^2 \vee \psi(XY_1Y_2) \big)$$

  – on vertex-colored paths, parameterized by formula size

- **NP**-hard to find exact answers for:

$$\psi_1(X) \wedge \forall Y_1 \forall Y_2 \forall Y_3 \forall Y_4 \big((|Y_1| \neq |Y_2|) \vee (|Y_3| = |Y_4|) \vee \psi_2(XY_3Y_4) \big),$$

  – on vertex-colored bounded-depth trees

Taking a Step Back

$$\varphi(\underbrace{X_1\ X_2\ \ \dots\ \ ) = \forall Y_1\ \forall Y_2\ \ \ \dots}_{\text{often used for comparisons}}\ \ \underbrace{\exists Z_1\ \forall Z_2\ \ \ \dots}_{\text{usually not used for comparisons}}$$

$$\varphi(\underbrace{X_1\ X_2\ \ \dots}_{\substack{\text{comparisons allowed}\\ \text{(first estimate)}}}\ ) = \underbrace{\forall Y_1\ \forall Y_2\ \ \ \dots\ \ \ \ \exists Z_1\ \forall Z_2\ \ \ \dots}_{\substack{\text{comparisons forbidden}\\ \text{(first estimate)}}}$$

- Restricting comparisons only to the first quantifier group would only capture some problems
- Restricting comparisons to the first two quantifier groups is too much

- Solution: only limited comparisons in the 2nd group

(Blocked) $\boxed{\forall}$CMSO Formulas

- $\boxed{\forall}$CMSO$_{1/2}$ logic contains all formulas that are built from *blocks* via ∃, v, ∧

- Block: CMSO[$\lessgtr$] formula with restricted comparisons:
  - Only non-negative coefficients and negation-normalized
  - At most one weight comparison has a weight term which involves variables in the 1st two quantifier groups
  - All other weight comparisons are exclusively for the first quantifier group

Meta-Theorem for Treewidth

What is this?

Theorem 11 (Approximation of $\boxed{\forall}\mathrm{CMSO}_2$). *Given a $\boxed{\forall}\mathrm{CMSO}_2$-query $(\varphi, \hat{t})$, an accuracy $0 < \varepsilon \leq 0.5$ and a graph G with matching signature, we can compute a $(1+\varepsilon)$-approximate answer to $(\varphi, \hat{t})$ on G in time at most*

$$\left(\frac{\log(M)}{\varepsilon}\right)^{f_1(|\varphi|, \mathrm{tw}(G))} \cdot |V(G)|^{1.001},$$

useful for exponential numbers encoded in binary

and in particular in time at most

$$\left(\frac{1}{\varepsilon}\right)^{f_2(|\varphi|, \mathrm{tw}(G))} \cdot M^{0.001} \cdot |V(G)|^{1.001}$$

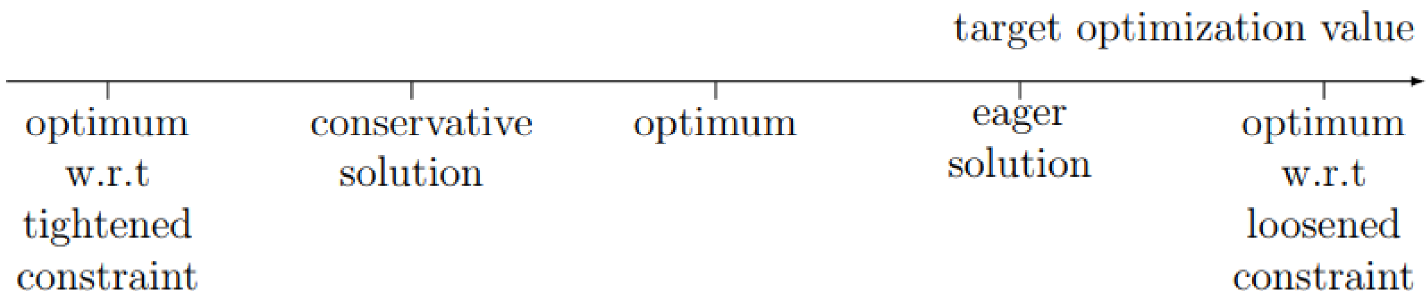
useful for unary-encoded numbers

where f_1, f_2 are computable functions and M is two plus the highest number that occurs in any weight term of φ or as any weight in G.

Approximate Solutions

- Original constraint: $|X| \leq |Y|$
  - Target value (=optimum): 0, -∞, 50, …
- Loosened constraint: $|X| \leq (1 + \varepsilon) \cdot |Y|$
  - Optimum could increase but not decrease
- Tightened constraint: $(1 + \varepsilon) \cdot |X| \leq |Y|$
  - Optimum could decrease but not increase
- A (1+ε)-approximate solution contains two solutions:
  – An eager one that is at least as good as the original optimum while satisfying the loosened constraints
  – A conservative one that satisfies the original constraints while being at least as good as the optimum under the tightened constraints

Approximate Solutions



- A (1+ε)-approximate solution contains two solutions:
  - An eager one that is at least as good as the original optimum while satisfying the loosened constraints
  - A conservative one that satisfies the original constraints while being at least as good as the optimum under the tightened constraints

Meta-Theorem for Cliquewidth

Theorem 10 (Approximation of $\boxed{\forall}$CMSO$_1$). *Given a* $\boxed{\forall}$CMSO$_1$*-query* $(\varphi, \hat{t})$*, an accuracy* $0 < \varepsilon \leq 0.5$ *and a graph* G *with matching signature, we can compute a* $(1+\varepsilon)$*-approximate answer to* $(\varphi, \hat{t})$ *on* G *in time at most*

$$\left(\frac{\log(M)}{\varepsilon}\right)^{f_1(|\varphi|, \mathrm{cw}(G))} \cdot |V(G)|^2,$$

useful for exponential numbers encoded in binary

and in particular in time at most

$$\left(\frac{1}{\varepsilon}\right)^{f_2(|\varphi|, \mathrm{cw}(G))} \cdot |V(G)|^2 \cdot M^{0.001}$$

useful for unary-encoded numbers

where f_1*,* f_2 *are computable functions and* M *is two plus the highest number that occurs in any weight term of* φ *or as any weight in* G*.*

– Notice that we have a lot of freedom in choosing ε
– What if we set ε to be so small that the loosened constraints are integer-equivalent to the original ones?

Exact Meta-Theorem

Theorem 12 (Evaluating Queries Exactly). *Given a* $\boxed{\forall}\mathrm{CMSO}_1$*-query (or* $\boxed{\forall}\mathrm{CMSO}_2$*-query)* $(\varphi, \hat{t})$ *and a graph* G *with matching signature and cliquewidth (or treewidth) at most* k*, we can compute an answer to* $(\varphi, \hat{t})$ *on* G *in time*

$$(M + |V(G)|)^{f(|\varphi|,k)}$$

where f *is some computable function and* M *is the highest number that occurs in any weight term of* φ *or as any weight in* G*.*

- Encoding a problem as a $\boxed{\forall}\mathrm{CMSO}_{1/2}$ query yields **XP**-tractability and **FPT**-time computation of “approximate” (eager/conservative) solutions

Some Examples

- **Equitable k-Coloring / k-Connected Partition**
  - $\forall$CMSO$_1$ formula whose size depends on k
  - When parameterizing by treewidth, we can also do:

$$
\begin{aligned}
\varphi_{\text{ECP-2}}(X) := &\Big(\forall Y : (\text{“}G[Y] \text{ is not a connected component of } G - X\text{”}) \vee |Y| \geq \left\lfloor \frac{n}{k} \right\rfloor\Big) \\
&\wedge \Big(\forall Y : (\text{“G[Y] is not a connected component of } G - X\text{”}) \vee |Y| \leq \left\lceil \frac{n}{k} \right\rceil\Big) \\
&\wedge \Big(\forall Y : (\text{“}Y \text{ does not contain precisely one vertex from each} \\
&\qquad\qquad \text{connected component of } G - X\text{”}) \vee |Y| \leq k\Big) \\
&\wedge \Big(\forall Y : (\text{“}Y \text{ does not contain precisely one vertex from each} \\
&\qquad\qquad \text{connected component of } G - X\text{”}) \vee |Y| \geq k\Big),
\end{aligned}
$$

where X is an edge set variable, Y is a vertex set variable and $n = |V(G)|$.

Some Examples

- **Equitable k-Coloring / k-Connected Partition**
  - $\forall$CMSO$_1$ formula whose size depends on k
- **Bounded-Degree Vertex Deletion**

$$\varphi_{\mathrm{BDVD}}(X) := \forall A : \big(\neg(\text{“}X \cap A = \emptyset\text{”} \ \wedge \ \text{“}A \text{ contains an } A\text{-universal vertex”})\big) \vee |A| \leq p+1.$$

- **Graph Motif**

$$\begin{aligned}\varphi_{\mathrm{GM}}(X) := \exists X_1, \ldots, X_k : &(\text{“}X \text{ is connected”}) \\ &\wedge (\text{“}X_1, \ldots, X_k \text{ is the partitioning of } X \text{ into colors } 1, \ldots, k, \text{ respectively”}) \\ &\wedge \Big(\bigwedge_{i \in [k]} |X_i| \leq M(i) \wedge M(i) \leq |X_i|\Big).\end{aligned}$$

Some Examples

- **Capacitated Vertex Cover / Capacitated Dom. Set**
  - Graph is preprocessed so that we can capture an "assignment" by the formula
- **Kidney Exchange**
  - Find a maximum-weight collection of vertex-disjoint cycles, each of length at most some given bound p
- **Non-graph problems**
  - Can capture known dynamic programming approximation algorithms for **Subset Sum** (incl. multidimensional variant), **Knapsack**, and other number problems via vertex weights

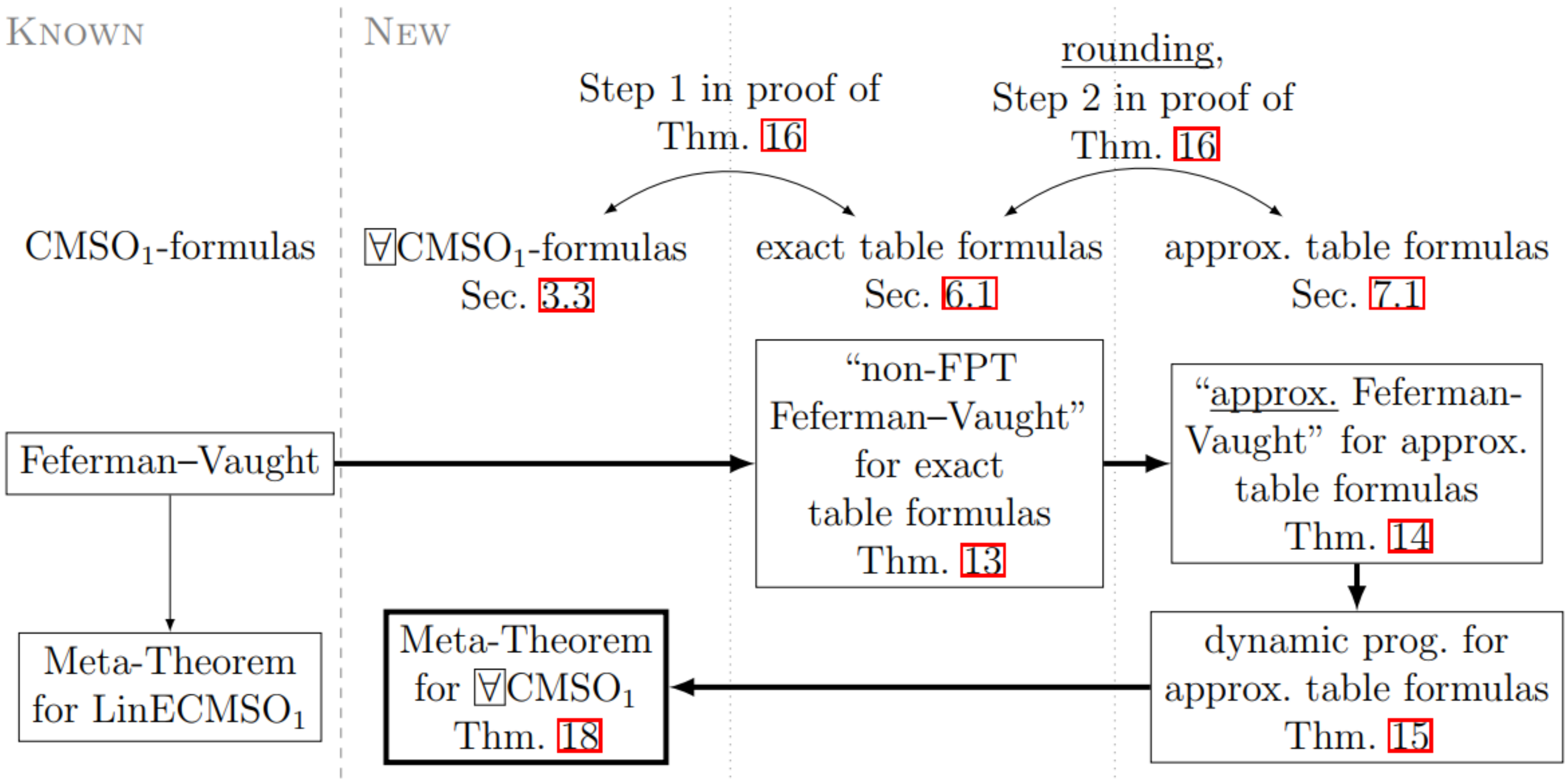
Last Examples

- **Max-Cut** and **Edge Dominating Set**
  - parameterized by cliquewidth
    - i.e., without edge set quantification

  - requires heavy preprocessing of input graph
  - probably harder to encode than to solve directly
  - only works for exact solutions (**XP**), not approximation
  - …but still unexpected

Thank you for your attention

Here is a “mindmap” for the proof



ALGORITHMS AND COMPLEXITY GROUP