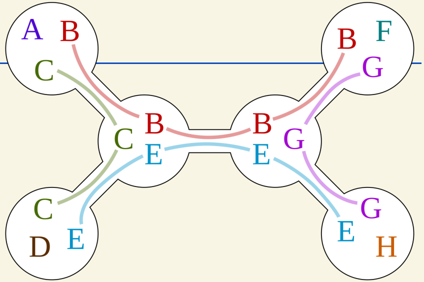
Treewidth III: Courcelle's Theorem



DM898: Parameterized Algorithms
Lars Rohwedder

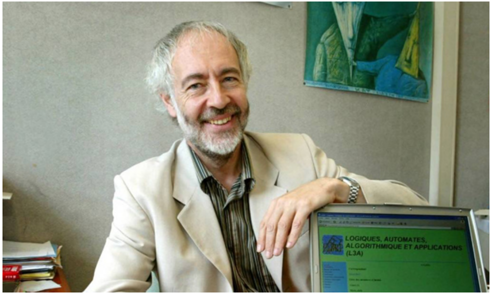
Today's lecture

- Monadic second order logic
- Courcelle's Theorem
- Optimization variant of Courcelle's Theorem

Context

- On bounded treewidth graphs, problems can often be solved by dynamic programming
- For what kind of problems does this work?

Informally: **Courcelle's Theorem** gives an FPT algorithm in treewidth for every problem that can be written in the very expressive monadic second order logic (which we introduce soon)



Bruno Courcelle

Source: `https://www.labri.fr/perso/courcell/ActSci.html`

Monadic second order logic

Informal overview

Monadic second order logic on graphs (MSO_2) is a formalism to describe properties of a graph that we are interested in checking.

Example:

$$\mathrm{3colorability} = \exists_{X_1, X_2, X_3 \subseteq V} \mathrm{partition}(X_1, X_2, X_3) \wedge \mathrm{indp}(X_1) \wedge \mathrm{indp}(X_2) \wedge \mathrm{indp}(X_3)\ , \text{ where}$$

$$\begin{aligned}\mathrm{partition}(X_1, X_2, X_3) = \forall_{v \in V} \Big[&(v \in X_1 \wedge v \notin X_2 \wedge v \notin X_3)\\ &\vee (v \notin X_1 \wedge v \in X_2 \wedge v \notin X_3)\\ &\vee (v \notin X_1 \wedge v \notin X_2 \wedge v \in X_3)\Big]\end{aligned}$$

$$\mathrm{indp}(X) = \forall_{u,v \in X} \neg \mathrm{adj}(u, v)$$

Formula and variables

A MSO_2 formula can have **variables** of one of the types

- single vertex: $v \in V$
- single edge: $e \in E$
- vertex set: $U \subseteq V$
- edge set: $F \subseteq E$

The **free** variables $x_1, \ldots, x_k$ of a formula ϕ (if there are any) are written in parantheses after the function name:

$$\phi(x_1, \ldots, x_k) = \cdots$$

We need to specify the type of the variables if it is not clear from the context.

A formula is evaluated on a given graph and specific values for the free variables. It evaluates to either **true** or **false**. We say that a graph G equipped with values for the free variables **satisfies** a formular if it evaluates to true

Constructions of formulas

Atomic formulas

- If v is a single vertex variable and U a vertex set variable, then the following is a MSO_2 formula: $v \in U$
- If e is a single edge variable and F a edge set variable, then the following is a MSO_2 formula: $e \in F$
- If x and y are variables of the same type, then the following is a MSO_2 formula: $x = y$
- If v is a single vertex variable and e is a single edge variable, then the following is a MSO_2 formula: $\mathrm{inc}(v, e)$ (evaluates to true if and only if v is incident to e)

Constructions of formulas

Atomic formulas

- If v is a single vertex variable and U a vertex set variable, then the following is a MSO_2 formula: $v \in U$
- If e is a single edge variable and F a edge set variable, then the following is a MSO_2 formula: $e \in F$
- If x and y are variables of the same type, then the following is a MSO_2 formula: $x = y$
- If v is a single vertex variable and e is a single edge variable, then the following is a MSO_2 formula: $\mathrm{inc}(v, e)$ (evaluates to true if and only if v is incident to e)

Logical operators

Let $\phi_1(x_1, \ldots, x_k), \phi_2(x_1, \ldots, x_k)$ be two MSO_2 formulas with the same free variables $x_1, \ldots, x_k$. Then the following are also MSO_2 formulas:

- $\neg\phi_1(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \wedge \phi_2(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \vee \phi_2(x_1, \ldots, x_k)$

Constructions of formulas

Atomic formulas

- If v is a single vertex variable and U a vertex set variable, then the following is a MSO_2 formula: $v \in U$
- If e is a single edge variable and F a edge set variable, then the following is a MSO_2 formula: $e \in F$
- If x and y are variables of the same type, then the following is a MSO_2 formula: $x = y$
- If v is a single vertex variable and e is a single edge variable, then the following is a MSO_2 formula: $\mathrm{inc}(v, e)$ (evaluates to true if and only if v is incident to e)

Logical operators

Let $\phi_1(x_1, \ldots, x_k), \phi_2(x_1, \ldots, x_k)$ be two MSO_2 formulas with the same free variables $x_1, \ldots, x_k$. Then the following are also MSO_2 formulas:

- $\neg\phi_1(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \wedge \phi_2(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \vee \phi_2(x_1, \ldots, x_k)$

Quantification

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula where x_i is a free single vertex variable. Then

- $\forall_{x_i \in V} \phi(x_1, \ldots, x_k)$ is a MSO_2 formula with free variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$
- $\exists_{x_i \in V} \phi(x_1, \ldots, x_k)$ is a MSO_2 formula with free variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$

We say that x_i is **bounded**. Same construction for single edge, vertex set, and edge set variables

Constructions of formulas

Atomic formulas

- If v is a single vertex variable and U a vertex set variable, then the following is a MSO_2 formula: $v \in U$
- If e is a single edge variable and F a edge set variable, then the following is a MSO_2 formula: $e \in F$
- If x and y are variables of the same type, then the following is a MSO_2 formula: $x = y$
- If v is a single vertex variable and e is a single edge variable, then the following is a MSO_2 formula: $\mathrm{inc}(v, e)$ (evaluates to true if and only if v is incident to e)

Logical operators

Let $\phi_1(x_1, \ldots, x_k), \phi_2(x_1, \ldots, x_k)$ be two MSO_2 formulas with the same free variables $x_1, \ldots, x_k$. Then the following are also MSO_2 formulas:

- $\neg\phi_1(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \wedge \phi_2(x_1, \ldots, x_k)$
- $\phi_1(x_1, \ldots, x_k) \vee \phi_2(x_1, \ldots, x_k)$

Quantification

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula where x_i is a free single vertex variable. Then

- $\forall_{x_i \in V} \phi(x_1, \ldots, x_k)$ is a MSO_2 formula with free variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$
- $\exists_{x_i \in V} \phi(x_1, \ldots, x_k)$ is a MSO_2 formula with free variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$

We say that x_i is **bounded**. Same construction for single edge, vertex set, and edge set variables

The class of MSO_2 formulas are exactly those that can be constructed using the rules above

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for ...
- write $x \neq y$ for ...
- write $\mathrm{adj}(u, v)$ for ...
- write $\exists_{v \in U} \phi(\ldots, v, \ldots)$ for ...
- write $X \subseteq Y$ for ...
- write $\phi_1(x_1, \ldots, x_k) \Rightarrow \phi_2(x_1, \ldots, x_k)$ for ...

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for $\neg(x \in X)$
- write $x \neq y$ for $\neg(x = y)$
- write $\mathrm{adj}(u, v)$ for ...
- write $\exists_{v \in U} \phi(\dots, v, \dots)$ for ...
- write $X \subseteq Y$ for ...
- write $\phi_1(x_1, \dots, x_k) \Rightarrow \phi_2(x_1, \dots, x_k)$ for ...

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for $\neg(x \in X)$
- write $x \neq y$ for $\neg(x = y)$
- write $\mathrm{adj}(u, v)$ for $(u \neq v) \wedge \exists_{e \in E} \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$
- write $\exists_{v \in U} \phi(\ldots, v, \ldots)$ for ...
- write $X \subseteq Y$ for ...
- write $\phi_1(x_1, \ldots, x_k) \Rightarrow \phi_2(x_1, \ldots, x_k)$ for ...

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for $\neg(x \in X)$
- write $x \neq y$ for $\neg(x = y)$
- write $\mathrm{adj}(u, v)$ for $(u \neq v) \wedge \exists_{e \in E} \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$
- write $\exists_{v \in U} \phi(\ldots, v, \ldots)$ for $\exists_{v \in V} [(v \in U) \wedge \phi(\ldots, v, \ldots)]$, same for $\forall$ and quantification over edges
- write $X \subseteq Y$ for ...
- write $\phi_1(x_1, \ldots, x_k) \Rightarrow \phi_2(x_1, \ldots, x_k)$ for ...

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for $\neg(x \in X)$
- write $x \neq y$ for $\neg(x = y)$
- write $\mathrm{adj}(u, v)$ for $(u \neq v) \wedge \exists_{e \in E} \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$
- write $\exists_{v \in U} \phi(\ldots, v, \ldots)$ for $\exists_{v \in V} [(v \in U) \wedge \phi(\ldots, v, \ldots)]$, same for $\forall$ and quantification over edges
- write $X \subseteq Y$ for $\forall_{x \in X} x \in Y$
- write $\phi_1(x_1, \ldots, x_k) \Rightarrow \phi_2(x_1, \ldots, x_k)$ for $\ldots$

Syntactic sugar

The following does not add to the expressive power of MSO_2, but simplifies notation:

- write $x \notin X$ for $\neg(x \in X)$
- write $x \neq y$ for $\neg(x = y)$
- write $\mathrm{adj}(u, v)$ for $(u \neq v) \wedge \exists_{e \in E} \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$
- write $\exists_{v \in U} \phi(\ldots, v, \ldots)$ for $\exists_{v \in V} [(v \in U) \wedge \phi(\ldots, v, \ldots)]$, same for $\forall$ and quantification over edges
- write $X \subseteq Y$ for $\forall_{x \in X} x \in Y$
- write $\phi_1(x_1, \ldots, x_k) \Rightarrow \phi_2(x_1, \ldots, x_k)$ for $\neg\phi_1(x_1, \ldots, x_k) \vee \phi_2(x_1, \ldots, x_k)$

Courcelle's Theorem

Courcelle's Theorem

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula with free variables $x_1, \ldots, x_k$. We can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ check if G satisfies ϕ for given values of $x_1, \ldots, x_k$, where f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Courcelle's Theorem

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula with free variables $x_1, \ldots, x_k$. We can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ check if G satisfies ϕ for given values of $x_1, \ldots, x_k$, where f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example (blackboard): Hamiltonicity is FPT in treewidth

Courcelle's Theorem

Let $\phi(x_1, \dots, x_k)$ be a MSO_2 formula with free variables $x_1, \dots, x_k$. We can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ check if G satisfies ϕ for given values of $x_1, \dots, x_k$, where f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example (blackboard): Hamiltonicity is FPT in treewidth

Courcelle's Theorem (optimization variant)

Let $\phi(Y_1, \dots, Y_p, x_1, \dots, x_k)$ be a MSO_2 formula with free edge/vertex set variables $Y_1, \dots, Y_p$ and free variables $x_1, \dots, x_k$ of any type. Let $c_1, \dots, c_p \in \mathbb{Z}$. For given values of $x_1, \dots, x_k$, we can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ find values for $Y_1, \dots, Y_p$ that satisfy $\phi(Y_1, \dots, Y_p, x_1, \dots, x_k)$ on G, if any such values exist, and maximize (or minimize)

$$c_1|Y_1| + c_2|Y_2| + \dots + c_p|Y_p| \ .$$

Here, f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Courcelle's Theorem

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula with free variables $x_1, \ldots, x_k$. We can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ check if G satisfies ϕ for given values of $x_1, \ldots, x_k$, where f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example (blackboard): Hamiltonicity is FPT in treewidth

Courcelle's Theorem (optimization variant)

Let $\phi(Y_1, \ldots, Y_p, x_1, \ldots, x_k)$ be a MSO_2 formula with free edge/vertex set variables $Y_1, \ldots, Y_p$ and free variables $x_1, \ldots, x_k$ of any type. Let $c_1, \ldots, c_p \in \mathbb{Z}$. For given values of $x_1, \ldots, x_k$, we can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ find values for $Y_1, \ldots, Y_p$ that satisfy $\phi(Y_1, \ldots, Y_p, x_1, \ldots, x_k)$ on G, if any such values exist, and maximize (or minimize)

$$c_1|Y_1| + c_2|Y_2| + \cdots + c_p|Y_p| \ .$$

Here, f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example 2 (blackboard): Vertex Cover is FPT in treewidth

Example 3 (blackboard): (Unweighted) Independent Set is FPT in treewidth

Example 4 (blackboard): Steiner Tree is FPT in treewidth

Courcelle's Theorem

Let $\phi(x_1, \ldots, x_k)$ be a MSO_2 formula with free variables $x_1, \ldots, x_k$. We can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ check if G satisfies ϕ for given values of $x_1, \ldots, x_k$, where f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example (blackboard): Hamiltonicity is FPT in treewidth

Courcelle's Theorem (optimization variant)

Let $\phi(Y_1, \ldots, Y_p, x_1, \ldots, x_k)$ be a MSO_2 formula with free edge/vertex set variables $Y_1, \ldots, Y_p$ and free variables $x_1, \ldots, x_k$ of any type. Let $c_1, \ldots, c_p \in \mathbb{Z}$. For given values of $x_1, \ldots, x_k$, we can in time $f(\|\phi\|, \mathrm{tw}(G)) \cdot n$ find values for $Y_1, \ldots, Y_p$ that satisfy $\phi(Y_1, \ldots, Y_p, x_1, \ldots, x_k)$ on G, if any such values exist, and maximize (or minimize)

$$c_1|Y_1| + c_2|Y_2| + \cdots + c_p|Y_p| \ .$$

Here, f is some computable function and $\|\phi\|$ is the encoding length of ϕ.

Example 2 (blackboard): Vertex Cover is FPT in treewidth

Example 3 (blackboard): (Unweighted) Independent Set is FPT in treewidth

Example 4 (blackboard): Steiner Tree is FPT in treewidth

We do not provide a proof of Courcelle's Theorem in this course

Limitations

- The parameter dependence f in Courcelle's theorem is astronomically large. Running time **cannot** even be bounded by

$$\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathrm{tw}(G)+\|\phi\|}}}}}}}_{\text{finite tower}} n \ .$$

  Explicit dynamic programs usually have a much better running time
- MSO_2 does not capture every property that can be checked in FPT time in treewidth. For example, arithmetics (including counting) usually cannot be done in MSO_2. Extensions exist that allow, for example, parity checks of sets (useful for example in Order Picking), but these extensions are still very limited