CS 5003: Parameterized Algorithms

Lecture 40

Krithika Ramaswamy

IIT Palakkad

Reference Books: Parameterized Algorithms by Cygan et al. and Kernelization by Fomin et al.

Problems Parameterized by Treewidth

Theorem 7.9. *Let G be an n-vertex graph given together with its tree decomposition of width at most k. Then in G one can solve*

- Vertex Cover *and* Independent Set *in time* $2^k \cdot k^{\mathcal{O}(1)} \cdot n$,
- Dominating Set *in time* $4^k \cdot k^{\mathcal{O}(1)} \cdot n$,
- Odd Cycle Transversal *in time* $3^k \cdot k^{\mathcal{O}(1)} \cdot n$,
- MaxCut *in time* $2^k \cdot k^{\mathcal{O}(1)} \cdot n$,
- q-Coloring *in time* $q^k \cdot k^{\mathcal{O}(1)} \cdot n$.

Theorem 7.10. *Let G be an n-vertex graph given together with its tree decomposition of width at most k. Then one can solve each of the following problems in G in time $k^{\mathcal{O}(k)} \cdot n$:*

- Steiner Tree,
- Feedback Vertex Set,
- Hamiltonian Path *and* Longest Path,
- Hamiltonian Cycle *and* Longest Cycle,
- Chromatic Number,
- Cycle Packing,
- Connected Vertex Cover,
- Connected Dominating Set,
- Connected Feedback Vertex Set.

Image Source: Parameterized Algorithms by Cygan et al., Springer, 2015

Monadic Second Order Logic on Graphs

MSO_2

* A formal language for expressing properties of graphs and objects inside these graphs like vertices, edges, and subsets of vertices/edges
* A formula of MSO_2 is a string following certain rules. It consists of the following:
  * Variables for: single vertices, single edges, subsets of vertices, subsets of edges
  * Logical connectives: $\vee$, $\wedge$, $=$, $\rightarrow$, $\neg$
  * Quantifiers $\exists, \forall$ over vertex/edge variables
  * Quantifiers $\exists, \forall$ over vertex/edge set variables
  * $\in$, $\subseteq$ for vertex/edge sets
  * May use $\neq$ and $\notin$ with conventional semantics

Monadic Second Order Logic on Graphs

MSO_2 Atomic Formulas

- $v \in X$ where v is a vertex (or edge) variable and X is a vertex (or edge) set variable
  - **Semantics:** the formula $v \in X$ is true iff the vertex corresponding to v is in the set corresponding to X in G
- $x = y$ where x and y are variables of the same type
  - **Semantics:** the formula $x=y$ is true iff the vertex/edge/set corresponding to x is same as the vertex/edge/set corresponding to y in G
- $X \subseteq Y$ where X and Y are vertex (or edge) set variables
  - **Semantics:** the formula $X \subseteq Y$ is true iff the set corresponding to X is contained in the set corresponding to Y

Monadic Second Order Logic on Graphs

MSO_2 Atomic Formulas (contd.)

* inc(v, e) where v is a vertex variable and e is an edge variable
    * Semantics: the formula inc(v, e) is true iff the vertex corresponding to v is an endpoint of the edge corresponding to e in G
* adj(v, u) where u and v are vertex variables
    * Semantics: the formula adj(u, v) is true iff the vertex corresponding to v is adjacent to the vertex corresponding to u

MSO_2 Formulas

* Constructed inductively from atomic formulas
    * If ϕ is a formula then $\neg\phi$, $\forall\phi$ and $\exists\phi$ are formulas
    * If ϕ_1, ϕ_2 are formulas then $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$ and $\phi_1 \implies \phi_2$ are formulas

Examples of MSO_2 Formulas

* The formula

  $$\exists C \subseteq V, \exists v_0 \in C, \forall v \in C, \exists u_1, u_2 \in C \ (u_1 \neq u_2 \wedge adj(u_1, v) \wedge adj(u_2, v))$$

  is true iff G has a cycle.

* The formula

  $$\exists C_1, C_2, C_3 \subseteq V \ (\forall v \in V \ (v \in C_1 \vee v \in C_2 \vee v \in C_3) \wedge (\forall u, v \in V \ adj(u, v) \implies$$
  $$(\neg(u \in C_1 \wedge v \in C_1) \wedge \neg(u \in C_2 \wedge v \in C_2) \wedge \neg(u \in C_3 \wedge v \in C_3))$$

  is true iff G is 3-colorable.

* The formula

  $$\exists X \subseteq V \ (\exists x \in X \wedge \exists y \notin X \wedge \forall x, y \in V \ (adj(x, y) \implies (x \in X \Leftrightarrow y \in X)))$$

  is true iff G is not connected.

Courcelle's Theorem

Theorem: If a graph property can be expressed as an MSO_2 formula ϕ, then there is an algorithm that given a graph G and a tree decomposition T of G, determines if G satisfies this property or not in $f(|\phi|, w(T))$ time for some computable function f.

For simple problems |phi| is a const and thus FPT wrt w(T) o/w use optimised

* f can be very large (double, triple exponential) and a direct DP algorithm can be more efficient
* If we can express a property in MSO_2, then we immediately infer that testing this property is FPT parameterized by the treewidth w of the input graph.
  * Existence of a vertex cover of size at most k
    * Uses an optimization version of Courcelle's theorem
  * Existence of a Hamiltonian cycle