Lecture notes for Apr 24, 2023 Tree-decomposition, Logic and Courcelle's theorem

Chun-Hung Liu

April 24, 2023

Recall that we show the following theorem last time.

Theorem 1 *Let w be a positive integer. Given a graph G and a tree-decomposition $(T, \mathcal{X})$ of G with width at most w, we can find a maximum stable set of G in time $O(w4^w|V(T)|)$.*

1 Finding a good tree-decomposition

According to Theorem 1, we want to, given an input graph G, find a tree-decomposition $(T, \mathcal{X})$ with small width and small $|V(T)|$. Ideally, we hope the width equals the tree-width. However, it is NP-hard to decide the tree-width.

Theorem 2 (Arnborg, Corneil, Proskurowski) *It is NP-hard to decide that given a graph G and a positive integer w, whether the tree-width of G is at most w.*

Note that the above problem is NP-hard when w is part of the input. As our purpose is to design an FPT algorithm with parameterization by the tree-width, the w is fixed for our applications. So the following theorem is good enough for us.

Theorem 3 (Bodlaender) *For every positive integer w, there is a linear time algorithm that given a graph G, either finds a tree-decomposition of G*

with width at most w, or correctly concludes that the tree-width of G is greater than w.

Note that the above theorem runs in linear time, so the tree-decomposition $(T, \mathcal{X})$ that it produces satisfies $|V(T)| = O(|V(G)|)$. Therefore we obtain the following corollary.

Corollary 4 *For every positive integer w, there exists a linear time algorithm that given a graph G with tree-width at most w, find a maximum stable set in G. More precisely, a maximum stable set of G can be found in time $f(w)|V(G)|$ for some function f for graphs G with tree-width at most w.*

Proof. Let w be a fixed positive integer. Let G be an input graph with tree-width at most w. Then we can obtain a tree-decomposition $(T, \mathcal{X})$ with width at most w and $|V(T)| = O(|V(G)|)$ in time $O(g(w)|V(G)|)$ (for some function g) by Theorem 3. Then we can find a maximum stable set of G in time $O(w4^w|V(T)|)$. Hence the total running time is $O(f(w)|V(G)|)$, where $f(w) = g(w) + w4^w$. ■

Note that this argument works for a much general setting than just finding a maximum stable set. For example, we can decide the k-colorability for graphs with bounded tree-width in a very similar way.

Corollary 5 *For every any positive integers w and k, there exists a linear time algorithm that given a graph G with tree-width at most w, decide whether G is k-colorable.*

Proof. It suffices to modify the dynamic programming in Theorem 1 to work for k-colorability. It can be done by considering all possible k-colorings of X_r. ■

2 Graph properties and logic

We have seen that some problems can be solved in polynomial time for graphs with bounded tree-width. They actually work in a much more general setting, and we will discuss it in this section.

We will express a graph property by using terminologies in logic. For example, the property that "G contains a stable set with size 2" can be expressed as "G satisfies $\exists x_1 \exists x_2\ (x_1 \neq x_2) \wedge \neg R(x, y)$", where $R(x, y)$ means

that "x is adjacent to y". Note that different properties require logic expressions with different "strength". The main goal here is to show that if a property that can be expressed by a logic expression with "low strength", then a FPT algorithm exists as long as the graphs have nice properties (such as having bounded tree-width etc.).

We will give more formal definition for this concept. But we remark that the logic concepts presented in this section are somehow simplified and might be slightly different from the use in model theory.

2.1 First-order graph properties

We say that a graph property P *can be expressed in FO* if there exists a first-order sentence ϕ such that a graph G satisfies P if and only if G satisfies ϕ, where a first-order sentence is defined as follows:

- A *first-order variable* (or just called "variable" for short) is a vertex of a graph.
- An *atomic formula* is either
  - "$x = y$", where x and y are variables, or
  - "$x \sim y$", which means "x is adjacent to y", or
  - "True" or "False",
- A *first-order formula* is defined by applying the following rules a finite number of times:
  - Every atomic formula is a first-order formula.
  - If ϕ_1 and ϕ_2 are formulas, then $\neg\phi_1$, $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$, $\exists x \phi_1$ and $\forall x \phi_1$ are also formulas.

    (Note that it implies that $\phi_1 \to \phi_2$ is also a formula because it is logically equivalent to $\neg\phi_1 \vee \phi_2$.)
- A variable is *free* in a first-order formula if it is not bounded by a quantifier (i.e. $\forall$ or $\exists$).
- A *first-order sentence* is a first-order formula with no free variables.

For simplicity, we write $\neg(x = y)$ as $x \neq y$, and write $\neg(x \sim y)$ as $x \not\sim y$.

Examples of properties expressible in FO:

1. For every positive integer k, the property "having a dominating set with size k" can be expressed as $\exists x_1 \exists x_2 ... \exists x_k \forall x \ \phi_1 \vee \phi_2 \vee ... \vee \phi_k$, where for every $i \in [k]$, ϕ_i is defined to be $(x = x_i) \vee (x \sim x_i)$.

2. For every positive integer k, the property "having a stable set with size k" can be expressed as $\exists x_1 \exists x_2 ... \exists x_k \ (\phi_{1,2} \wedge \phi_{1,3} \wedge ... \wedge \phi_{1,k}) \wedge (\phi_{2,3} \wedge \phi_{2,4} ... \wedge \phi_{2,k}) \wedge ... \wedge \phi_{k-1,k}$, where for every $1 \leq i < j \leq k$, $\phi_{i,j}$ is defined to be $x_i \not\sim x_j$.

3. For every graph H, the property "containing H as an induced subgraph" can be expressed as $\exists x_1 \exists x_2 ... \exists x_k \ (\phi_{1,2} \wedge \phi_{1,3} \wedge ... \wedge \phi_{1,k}) \wedge (\phi_{2,3} \wedge \phi_{2,4} ... \wedge \phi_{2,k}) \wedge ... \wedge \phi_{k-1,k}$, where for every $1 \leq i < j \leq k$, $\phi_{i,j}$ is defined to be $x_i \neq x_j \wedge \phi'_{i,j}$, and $\phi'_{i,j}$ is defined to be $x_i \sim x_j$ or $x_i \not\sim x_j$ (depending on whether the i-th vertex of H is adjacent to the j-th vertex).

4. For every graph H, the property "containing H as a subgraph" can be expressed in a similar way as "containing H as an induced subgraph" by removing $\phi_{i,j}$ for the pairs $(i.j)$ corresponding to non-adjacent vertices in H.

2.2 Monadic second-order logic

We say that a graph property P is a MSO_1-*property* if there exists a MSO_1-sentence ϕ expressible in MSO_1 such that a graph G satisfies P if and only if G satisfies ϕ, where a MSO_1-sentence is defined as follows:

- A *variable* is a vertex of a graph or a set of vertices.

  (We usually use letters of small case for vertices and letters of large case for set of vertices.)

- An *atomic formula* is either

  - "$x = y$", where x and y are variables, or
  - "$x \sim y$", which means "x is adjacent to y", or
  - "$x \in X$", which means "x is in the set X",

– "True" or "False",

- A MSO_1-*formula* is defined by applying the following rules a finite number of times:
  - Every atomic formula is a MSO_1-formula.
  - If ϕ_1 and ϕ_2 are formulas, then $\neg\phi_1$, $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$, $\exists x\phi_1$ and $\forall x\phi_1$ are also formulas.
- A MSO_1-*sentence* is a MSO_1-formula with no free variables.

Note that every FO-sentence is a MSO_1-sentence. But MSO_1-sentences are much stronger than FO-sentences.

Proposition 6 *For every positive integer k, "being k-colorable" is a MSO_1-property.*

Proof. It can be expressed as $\exists X_1 ... \exists X_k (\forall x \ \phi_0(x)) \wedge (\forall x \forall y \ \phi_1(x,y) \wedge \phi_2(x,y) \wedge ... \wedge \phi_k(x,y))$, where $\phi_0(x)$ is defined to be $x \in X_1 \vee x \in X_2 \vee ... \vee x \in X_k$, and for every $i \in [k]$, $\phi_i(x,y)$ is defined to be $((x \neq y) \wedge x \in X_i \wedge y \in X_i) \rightarrow x \not\sim y$. ■

Proposition 7 *"Being connected" is a MSO_1-property.*

Proof. "Being connected" is the negation of "has at least two components". So it suffices to express "has at least two components".

"X is a non-empty proper subset of $V(G)$", denoted by $\phi_0(X)$, can be expressed as $\exists x \exists y \ x \in X \wedge y \notin X$. "Has at least two components" can be expressed as $\exists X \ \phi_0(X) \wedge (\forall u \forall v (u \in X \wedge u \sim v) \rightarrow v \in X)$. ■

Let G and H be simple graphs. We say that H *is a minor of* G (or G *contains* H *as a minor*) if H can be obtained from G by repeatedly deleting vertices and edges and contracting edges. Note that it is equivalent to the statement that there exist pairwise disjoint subsets $X_1, X_2, ..., X_{|V(H)|}$ of $V(G)$ each inducing a connected subgraph such that for any $1 \leq i < j \leq |V(H)|$ if the i-th vertex of H is adjacent to the j-th vertex of H, then there exists an edge of G between X_i and X_j.

Proposition 8 *For every simple graph H, "containing H as a minor" is a MSO_1-property.*

Proof. Denote $V(H)$ by $h_1, h_2, ..., h_{|V(H)|}$. For every $1 \leq i < j \leq |V(H)|$, define $\varphi_{i,j}(X_i, X_j)$ to be $\forall x \forall y (x \in X_i \wedge y \in X_j) \rightarrow x \neq y$, which denotes "$X_i \cap X_j = \emptyset$". Let $\varphi = \wedge_{1 \leq i < j \leq |V(H)|} \varphi_{i,j}(X_i, X_j)$.

Denote $E(H)$ by $e_1, e_2 ..., e_{|E(H)|}$. For every $1 \leq i < j \leq |V(H)|$, if $h_i h_j \in E(H)$, then define $\phi_{i,j}(X_i, X_j)$ to be $\exists x \exists y \; x \in X_i \wedge y \in X_j \wedge x \sim y$. For every $k \in [|E(H)|]$, say $e_k = h_{a_k} h_{b_k}$, we define $\psi_k = \phi_{a_k, b_k}(X_{a_k}, X_{b_k})$.

By Proposition 7, for every $i \in [k]$, "$G[X_i]$ is connected", denoted by ϕ_i, can be expressed as a MSO_1-sentence. So "containing H as a minor" can be expressed as $\exists X_1 \exists X_2 ... \exists X_{|V(H)|} \varphi \wedge (\phi_1 \wedge \phi_2 \wedge .... \wedge \phi_k) \wedge (\psi_1 \wedge \psi_2 \wedge ... \wedge \psi_{|E(H)|})$. ■

Proposition 9 *"Being planar" is a MSO_1-property.*

Proof. "Being planar" is equivalent to "not containing K_5 as a minor and not containing $K_{3,3}$ as a minor" by Wagner's theorem. So this proposition follows from Proposition 8. ■

We say that a graph property P is a MSO_2-*property* if there exists a MSO_2-sentence ϕ expressible in MSO_2 such that a graph G satisfies P if and only if G satisfies ϕ, where a MSO_2-sentence is defined as follows:

- A *variable* is a vertex, edge, a set of vertices, or a set of edges.
- An *atomic formula* is either
  - "$x = y$", where x and y are variables, or
  - "$x \sim y$", which means "x is adjacent to y" (requiring both x and y are vertices), or
  - "$x \in X$", which means "x is in the set X", or
  - "$x \approx y$", which means "x and y are incident" (requiring one of x and y is a vertex and the other is an edge), or
  - "True" or "False",
- A MSO_2-*formula* is defined by applying the following rules a finite number of times:
  - Every atomic formula is a MSO_2-formula.
  - If ϕ_1 and ϕ_2 are formulas, then $\neg \phi_1$, $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$, $\exists x \phi_1$ and $\forall x \phi_1$ are also formulas.

- A MSO_2-*sentence* is a MSO_2-formula with no free variables.

Note that every MSO_1-sentence is a MSO_2-sentence. But MSO_2-sentences are much stronger than MSO_1-sentences.

Proposition 10 *"Having a Hamiltonian cycle" is a MSO_2-property.*

Proof. Let G be a graph. Note that a subset Y of $E(G)$ is the edge-set of a Hamiltonian cycle if and only if every vertex of G is incident with exactly two edges in Y and for every nonempty proper subset X of $V(G)$, there exists at least one edge in Y between X and $V(G) - X$.

For a set of edges Y and a vertex x, "there are exactly two edges in Y incident with x", denoted by $\phi_1(Y, x)$, can be expressed as $\exists e_1 \exists e_2 (e_1 \in Y \wedge e_2 \in Y \wedge e_1 \neq e_2 \wedge e_1 \approx x \wedge e_2 \approx x) \wedge \forall e_3 ((e_3 \in Y \wedge e_3 \neq e_1 \wedge e_3 \neq e_2) \rightarrow e_3 \not\approx x)$. For a set X of vertices, "X is a non-empty proper subset", denoted by $\phi_2(X)$, can be expressed as "$\exists x_1 \exists x_2\ x_1 \neq x_2 \wedge x_1 \in X \wedge x_2 \notin X$. For a set X of vertices and a set of edges Y, "there exists an edge in Y between X and $V(G) - X$", denoted by $\phi_3(X, Y)$, can be expressed as $\exists y \exists x_1 \exists x_2\ y \in Y \wedge x_1 \approx y \wedge x_2 \approx y \wedge x_1 \in X \wedge x_2 \notin X$.

So "having a Hamiltonian cycle" can be expressed as $\exists Y ((\forall x \phi_1(Y, x)) \wedge \forall X (\phi_2(X) \rightarrow \phi_3(X, Y)))$. ■

The following is a famous theorem. We will not give its proof here, but it somehow uses the ideas about dynamic programming that we mentioned before.

Theorem 11 (Courcelle) *For every graph MSO_2-property P, there exist a function f and an algorithm such that given a graph G with tree-width at most w, deciding whether G satisfies P can be done in $f(w)|V(G)|$ time.*

Courcelle's theorem is strong. But it is limited to graphs with bounded tree-width. Can we relax this condition, for example, by considering planar graphs instead? It is not possible unless NP=P, since 3-colorability is NP-hard for planar graphs and is a MSO_1-property. So if we want to relax the condition on graph structure, then we have to weaken the graph property. We will do it in the next lecture.