Courcelle's theorem

Nikos Tsilivis

National Technical University of Athens

ntsilivis96@gmail.com

January 26, 2020

Overview

Abstract

Theorem

Every graph property definable in the monadic second-order logic of graphs can be decided in linear time on graphs of bounded treewidth.

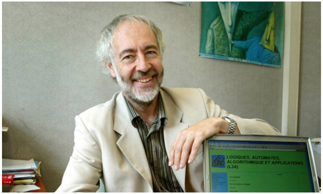


Figure: Bruno Courcelle, 2004

Tree Automata

We consider finite trees T with a root r_T, which have labels on their nodes over an alphabet Σ (Σ^{**} being the set of all of them).
We wish to extend the familiar(?) DFA to be able to parse trees.

Definition (Deterministic Finite bottom-up Tree Automata, in short *TA*)

A *TA* is a 5-tuple $(Q, \Sigma, \delta, S, F)$ and a number f, where:

- Q: a finite set of states
- Σ: a finite alphabet
- δ: a transition function $\bigcup_{1 \leq i \leq f} (\underbrace{Q \times Q \times ... \times Q}_{\text{i times}} \times \Sigma) \to Q$
- S: the set of initial states $S \subseteq Q$
- F: the set of final states $F \subseteq Q$

As we see from the definition, the transition between states is ruled by the state of current node's children and its label. To clarify the definition, let's see an example. (on board)

Example (Transition function)

States	Label	$\delta()$
q_0	0	q_0
q_0	1	q_1
q_1	0	q_1
q_1	1	q_1
q_0, q_0	0	q_0
q_0, q_0	1	q_1
q_0, q_1	0	q_0
q_0, q_1	1	q_1
q_1, q_0	0	q_0
q_1, q_0	1	q_0
q_1, q_1	0	q_0
q_1, q_1	1	q_1

Tree decomposition

Definition (Tree decomposition)

Let $G = (V, E)$ be a graph. A tree T is a **tree decomposition** of G, if the following are true:

- Every node t of T is a set $B_t \subseteq V$ of nodes of G (namely *bag*), with $\bigcup_t B_t = V$
- For each edge $e = (u, v) \in E$, there exists bag B_t such that $\{u, v\} \subseteq B_t$
- For each node $u \in G$, the induced subgraph of T consisting of all bags that include u is connected.

Treewidth

We define as *width* of a tree decomposition the size of the biggest bag minus 1.

Definition (Treewidth)

We call **treewidth** of a graph the minimum width across all of its tree decompositions. Formally,

$$tw(G) \hat{=} \min_{T:\ T \text{ tree decomposition of } G} \{\max_{t \in T}(|B_t| - 1)\}$$

This pair of concepts encapsulate the idea of "tree-likeness" of an arbitrary graph. For our needs, we need only to know how to find the treewidth of a graph. Sadly,..

Computational aspects of treewidth

Theorem (Arnborg, Corneil, Proskurowski 1987)

The following problem is NP-hard:
Given a graph G and a number k, does G have treewidth of size at most k?

But if we fix the *parameter* treewidth as a constant then we have the following result, which establishes the conditional tractability of the problem.

Theorem (Bodlaender, Kloks 1996)

Given a graph G, there exists an algorithm which returns a tree decomposition of G of width w (if there exists) in time $\mathcal{O}(2^{\mathcal{O}(w^3)} \cdot |G|)$.

The above method (to fix a parameter as a constant and attack the problem) lies at the heart of the field of *Parameterized Complexity*.

NP-Completeness

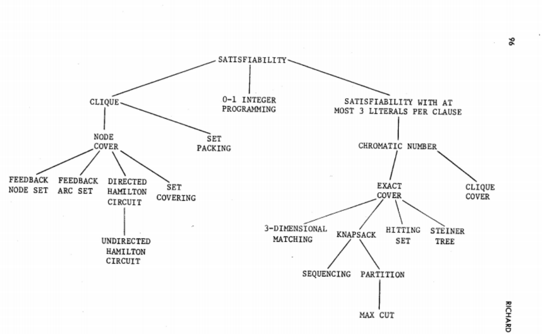


FIGURE 1 - Complete Problems

Beyond NP-Completeness - FPT complexity

Attacking NP-Complete problems by:

- Approximation Algorithms (sacrificing precision)
- Probabilistic Algorithms (sacrificing control)
- *Fixed Parameter Tractable* Algorithms (sacrificing a number of instances)

Example: Vertex Cover(n,k)

We fix the size of the vertex cover at a constant k. For every edge, one of its endpoints must belong to the cover. Doing this recursively for every edge, creates $O(2^k)$ subproblems and in every one of them, we can check if its nodes form a vertex cover in $O(n)$ time. Thus, the previous procedure yields a $O(2^k n)$ algorithm for the vertex cover.

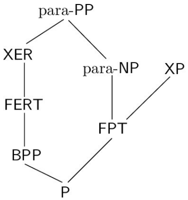
FPT Complexity

Definition (Parameterized problem)

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. The second component is called the parameter of the problem.

Definition (FPT Class)

A parameterized problem L is fixed-parameter tractable if the question "$(x, k) \in L$" can be decided in running time $f(k) \cdot |x|^{O(1)}$, where f is an arbitrary function depending only on k. The corresponding complexity class is called FPT.



Parse trees

We aim to connect the concepts of graphs of bounded treewidth and Tree Automata. We start with a couple of essential definitions.

Definition (t-boundaried graphs)

A **t-boundaried** graph is a graph G together with t distinguished vertices, which they get labels from $\{1, .., t\}$. We denote these vertices with $\partial(G)$ and they form the *boundary* of G.

Definition (graph gluing $\oplus$)

Let G_1, G_2 be t-boundaried graphs. We denote by $G_1 \oplus G_2$ the graph obtained by "gluing" G_1, G_2 at their boundary.

(example on board)

There is also a more flexible version of $\oplus$, namely the t-boundaried operator $\otimes$. These operators capture the core structure of bounded treewidth graphs, as the next theorem reveals.

Parsing theorem (1/2)

Theorem

There exists a linear algorithm which converts a tree decomposition of width t into a binary parse tree using $t+1$ parsing operators.

The proof uses the following 6 operators:

- $\varnothing$: creates $t+1$ vertices, with labels $1, .., t+1$.
- γ: shifts cyclically by 1 (modulo $t+1$) the labels of the boundary.
- i: interchanges the vertices labeled as $1, 2$.
- e: adds an edge between vertices $1, 2$.
- u: adds a vertex and labels it 1.
- $\oplus$.

Parsing theorem (2/2)

-And who cares? *you may ask*.
-The big deal here is that **for each bounded treewidth graph** we have now an alphabet (the 6 operators) and a tree (its corresponding parse tree) to feed as input in a suitable Tree Automaton and we are able to check if a graph has a structural property of interest.
-But Courcelle's theorem promises the decidability of a whole class of properties. Do we need to construct a TA for every one of them?
-Thank Göd, no!

Graph-theoretic Myhill -Nerode (1/2)

Definition (Graph Universes)

1. Large universe, U_t^{large}: Consists of all t-boundaried graphs.
2. Small universe, U_t^{small}: Consists of t-boundaried graphs obtained by parse trees.

Definition (Regular equivalence relation of graphs)

Let U be a universe of t-boundaried graphs, $G_1, G_2 \in U$ and F be a family of graphs in U, then we write $G_1 \sim_F G_2$ iff:

$$\forall H \in \mathcal{U}(G_1 \oplus H \in F \iff G_2 \oplus H \in F)$$

Definition (t-finite stae)

Let F be a family of graphs. We call F *t-finite state* when the parse trees corresponding to the set $F \cap U_t^{\text{small}}$ are finite state (that is, there exists a Tree Automaton that recognizes them)

Graph-theoretic Myhill-Nerode (2/2)

Reminder: The index of an equivalence relation is the number of its equivalence classes.
We are now ready to express the theorem:

Theorem

Let F be a family of graphs. Then the following are equivalent:

- *F is t-finite state*
- *the relation $\sim_F$ has finite index over U_t^{small}*

So, now, in order to show that there is an automaton which recognizes a family of graphs, we need only to show that the induced relation $\sim_F$ has finite index. This is the way we are going to prove Courcelle's theorem.
First of all, we need to define the logical language.

Monadic Second Order Logic

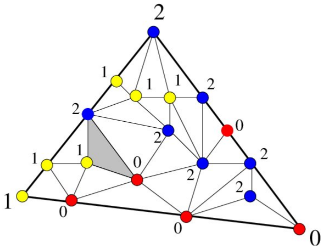
- A logical language for graphs
- Part of Second Order Logic (quantification over predicates of one variable -monadic- or equivalently sets)
- Syntax consists of:
  - Logical connectives $\wedge, \vee, \neg$
  - Variables for a vertex, an edge, a set of vertices and a set of edges
  - Quantifiers $\exists, \forall$ over variables
  - Five binary relations:
    - $u \in U$, where u is a vertex variable and U a vertex set variable
    - $d \in D$, where d is an edge variable and D an edge set variable
    - $inc(d, u)$, where d is an edge variable, u a vertex variable and the interpretation is that the edge d is incident on the vertex u
    - $adj(u, v)$, where both u, v are vertex variables and the interpretation is that u and v are adjacent vertices.
    - Equality relation $=$ over variables

MSO_2 Examples (1/2)

Example (3-colorability)

3-colorability $\equiv \exists X_1, X_2, X_3 \subseteq V(\text{part}(X_1, X_2, X_3) \wedge \text{indp}(X_1) \wedge \text{indp}(X_2) \wedge \text{indp}(X_3))$, where:

- $\text{part}(X_1, X_2, X_3) \equiv \forall v \in V((v \in X_1 \vee v \in X_2 \vee v \in X_3) \wedge \neg(v \in X_1 \wedge v \in X_2) \wedge \neg(v \in X_1 \wedge v \in X_3) \wedge \neg(v \in X_2 \wedge v \in X_3))$
- $\text{indp}(X) \equiv \forall u, v \in X \neg\text{adj}(u, v)$



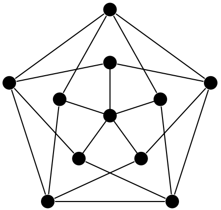
MSO_2 Examples (2/2)

Example (Hamilton Cycle)

$$\text{Hamiltonicity} \equiv \exists R \subseteq E(\text{conn}(R) \wedge \forall v \in V \text{deg2}(v, R)),$$

where:

- $\text{conn}(R) \equiv \forall Y \subseteq V[(\exists u \in V(u \in Y) \wedge \exists v \in V(v \notin Y)) \Rightarrow (\exists e \in R, u \in Y, v \notin Y(\text{inc}(u, e) \wedge \text{inc}(u, e)))]$
- $\text{deg2}(u, R) \equiv \exists e_1, e_2 \in R[(e_1 \neq e_2 \wedge \text{inc}(e_1, u) \wedge \text{inc}(e_2, u) \wedge (\forall e_3 \in R(\text{inc}(e_3, u) \Rightarrow (e_3 = e_1 \vee e_3 = e_2)))]$



proof of Courcelle's theorem(1/11)

Theorem

Let ϕ be a sentence defined in MSO_2 and F be the family of graphs which satisfy it. The relation $\sim_F$ has finite index in the U_t^{large}.

Proof.

In correspondence to the free variables $(fr(0), fr(1), Fr(0), Fr(1))$ of ϕ, we define a *partial equipment signature* σ, consisting of:

- Disjoint sets $int_0(\sigma), \partial_0(\sigma) \subseteq fr(0)$ and a map $f_\sigma^0 : \partial_0(\sigma) \to \{1, .., t\}$
- Disjoint sets $int_1(\sigma), \partial_1(\sigma) \subseteq fr(1)$ and a map $f_\sigma^1 : \partial_1(\sigma) \to \{1, .., t\}^2$
- A set $U_\sigma \subseteq \{1, ..t\}$ for each vertex set variable $U \in Fr(0)$
- A set $D_\sigma \subseteq \{1, ..t\}^2$ for each edge set variable $D \in Fr(1)$

Proof (Continue).

We say that a t-boundaried graph X is σ-**partially equipped**, if it has distinguished elements (vertices and edges) for each element of signature σ and the following properties hold:

- If u is a vertex variable and $u \in int_0(\sigma)$, then the corresponding vertex of X, u^X, must belong in the interior of X.
- If u is a vertex variable and $u \in \partial_0(\sigma)$, then u^X must be the only vertex satisfying $f(u^X) = f_\sigma^0(u)$.
- If d is an edge variable and $d \in int_1(\sigma)$, then edge d^X must have at least on of its endpoints in the interior of X.
- If d is an edge variable and $d \in \partial_1(X)$, then edge $d^X = (u^X, v^X)$ must satisfy $f_\sigma^1(d) = \{f(u^X), f(v^X)\}$.
- If U is a vertex set variable and $U \in Fr(0)$, then the corresponding vertex set U^X of X must satisfy $f(U^X \cap \partial(X)) = U_\sigma$.

Proof (Continue).

- If D is an edge set variable and $D \in Fr(1)$, then the corresponding edge set D^X of X must satisfy $\{\{f(u^X), f(v^X)\} : (u^X, v^X) \in E\} \cap \partial^2(X) = D_\sigma$.

We also define the complementary signature $\bar{\sigma}$ of a signature σ:

- $\partial_0(\bar{\sigma}) = \partial_0(\sigma)$ and $f^0_{\bar{\sigma}} = f^0_\sigma$
- $int_0(\bar{\sigma}) = fr(0) - \partial_0(\sigma) - int_0(\sigma)$
- $\partial_1(\bar{\sigma}) = \partial_1(\sigma)$ and $f^1_{\bar{\sigma}} = f^1_\sigma$
- $int_1(\bar{\sigma}) = fr(1) - \partial_1(\sigma) - int_1(\sigma)$
- For each vertex set variable $U \in Fr(0)$: $U_{\bar{\sigma}} = U_\sigma$
- For each edge set variable $D \in Fr(1)$: $D_{\bar{\sigma}} = D_\sigma$

proof of Courcelle's theorem (4/11)

Proof (Continue).

The previous odd definition simply assures that for graphs X, Z equipped with equipments $\sigma, \bar{\sigma}$ respectively, sentence ϕ has also consistent interpretation for graph $X \oplus Z$.

After these quite tedious definitions, we can state the main claim we seek to prove:

For each sentence $\phi \in MSO_2$, with equipment signature σ, the relation $\sim_\phi$ has finite index in the universe of t-boundaried σ-partially equipped graphs, where $X \sim_\phi$ Y when for each $\bar{\sigma}$-partially equipped t-boundaried graph Z holds:

$$X \oplus Z \models \phi \iff Y \oplus Z \models \phi$$

We will show the claim with the use of structural induction.

Proof (Continue).

- Atomic formulas:
  - $u \in U$
  - $d \in D$
  - $inc(e, u)$
  - $adj(u, v) \equiv \exists d \in D(inc(d, u) \wedge inc(d, v))$ ✓

All easy with case checking for different signatures.
e.g. $\phi = d \in D, \partial_1(\sigma) = d, D_\sigma = \emptyset$, then $\bar{\sigma} = \sigma$ and $\forall X, Z(X \oplus Z) \not\models \phi$, as $d \notin D$ (since D is limited to interior edges).
So index 1 (all equivalent).

Proof (Continue).

- Inductive Step. Formula ϕ must be one of the below:
  1. $\phi = \neg\phi^{'}$
  2. $\phi = \phi_1 \wedge \phi_2$
  3. $\phi = \exists u\phi^{'}$
  4. $\phi = \exists d\phi^{'}$
  5. $\phi = \exists U\phi^{'}$
  6. $\phi = \exists D\phi^{'}$,

  where $\phi^{'}, \phi_1, \phi_2$ satisfy the desired property, that is the induced relation has finite index.

Proof (Continue).

1. $\phi = \neg\phi^{'}$
   Let X, Y such that $X \sim_{\phi'} Y$, that is:

$$X \oplus Z \models \phi^{'} \iff Y \oplus Z \models \phi^{'} \; \forall Z$$

Suppose that X, Y are not $\sim_{\phi}$ equivalent, that is there exists Z:

$$X \oplus Z \models \phi \text{ and } Y \oplus Z \not\models \phi \iff$$

$$X \oplus Z \models \phi \text{ and } Y \oplus Z \models \neg\phi \iff$$

$$X \oplus Z \models \neg\phi^{'} \text{ and } Y \oplus Z \models \phi^{'},$$

a contradiction. We proved $X \sim_{\phi'} Y \Rightarrow X \sim_{\phi} Y$, so $\sim_{\phi'}$ is a refinement of $\sim_{\phi}$, and since $\sim_{\phi'}$ has finite index, so does $\sim_{\phi}$.

Proof (Continue).

2. $\phi = \phi_1 \wedge \phi_2$
   Define $\sim \equiv \sim_{\phi_1} \cap \sim_{\phi_2}$ (with appropriate signatures..). $\sim$ has finite index as the intersection of eq. relations of finite index.
   Let X, Y be graphs such that $X \sim Y$, that is $\forall Z$:

$$X \oplus Z \models \phi_1 \iff Y \oplus Z \models \phi_1$$

   and

$$X \oplus Z \models \phi_2 \iff Y \oplus Z \models \phi_2$$

   Suppose X, Y are not $\sim_\phi$ equivalent, then exists Z:

$$X \oplus Z \models \phi \wedge Y \oplus Z \models \neg\phi \iff (X \oplus Z \models \phi_1 \wedge \phi_2) \wedge (Y \oplus Z \models \neg\phi_1 \vee \neg\phi_2)$$

   So either $\sim_{\phi_1}$ or $\sim_{\phi_1}$ equivalence of X, Y breaks down, **a contradiction**. So, $\sim_\phi$ has finite index, in this case as well.

Proof (Continue).

3. $\phi = \exists u \phi^{'}$, where u is a vertex variable free in $\phi^{'}$
   Define $a(X, Z)$: exists vertex u^X in the interior of X such that $X_u \oplus Z \models \phi^{'}$, where X_u is graph X further equipped with vertex u^X (which corresponds to variable u). Based on that, define an extra relation:
   $$X \sim_{\exists u} Y : \forall Z[a(X, Z) \iff a(Y, Z)]$$
   The suitable, final, relation for this case is defined as $X \sim Y$ iff:
   1. $X \sim_{\phi^{'}} Y$
   2. $X_i \sim_{\phi^{'}} Y_i \forall i \in \{1, ..., t\}$, where X_i denotes the partially equipped t-boundaried graph X, further equipped with boundary vertex labeled i (corresponding to variable u), and
   3. $X \sim_{\exists u} Y$

   It has finite index.

proof of Courcelle's theorem (10/11)

Proof (Continue).

3. Let X, Y such that $X \sim Y$ and suppose that they are not $\sim_\phi$ equivalent, that is $\exists Z : X \oplus Z \models \phi$ and $Y \oplus Z \models \neg\phi$. If v is the vertex of $X \oplus Z$ corresponding to variable u, consider the cases:
   - u belongs in the interior of Z, then:
     $$X \oplus Z_u \models \phi^{'} \iff Y \oplus Z_u \models \phi^{'}, \text{ from (1)},$$
     meaning $Y \oplus Z \models \phi$, a contradiction.
   - u belongs in the boundary of X, then for a $i \in \{1, ..t\}$ it holds:
     $$X_i \oplus Z_i \models \phi^{'} \iff Y_i \oplus Z_i \models \phi^{'}, \text{ from (2)},$$
     meaning $Y \oplus Z \models \phi$, a contradiction.
   - u belongs in the interior of X, then from (3) there exists vertex $v^{'}$ in Y such that:
     $$X_u \oplus Z \models \phi^{'} \iff Y_u \oplus Z \models \phi^{'},$$
     meaning $Y \oplus Z \models \phi$, a contradiction.

proof of Courcelle's theorem (11/11)

Proof (Continue).

3. Thus, $\sim$ refines $\sim_\phi$ and $\sim_\phi$ has finite index.
4.
5.
6. Likewise for $(4), (5), (6)$ (define statements ensuring the equivalence in the interior of X, as in (3))

□

The proven theorem plus Bodlaender's algorithm establish the promised result.

References

Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer (2013)

Marek Cygan , Fedor V. Fomin , Lukasz Kowalik , Daniel Lokshtanov , Daniel Marx , Marcin Pilipczuk , Michal Pilipczuk , Saket Saurabh: *Parameterized Algorithms*, Springer Publishing Company, Incorporated, 2015

The End