Université Paris Cité – LIPADE

Algorithmic Complexity

Introduction to Advanced Concepts

Jean-Guy Mailly (jean-guy.mailly@u-paris.fr)

2022

Outline

Boolean Hierarchy

- Some problems cannot be easily classified in PH-classes
- The Boolean Hierarchy is useful to discriminate problems which are between the first (NP, coNP) and second (Σ_2^{P}, Π_2^P) levels of the polynomial hierarchy

Boolean Hierarchy

- Some problems cannot be easily classified in PH-classes
- The Boolean Hierarchy is useful to discriminate problems which are between the first (NP, coNP) and second (Σ_2^{P}, Π_2^P) levels of the polynomial hierarchy

The BH is the set of decision problems defined inductively by:

- $\mathrm{BH}_1 = \mathrm{NP}$
- $\mathrm{BH}_{2i} = \{\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2 \mid \mathcal{L}_1 \in \mathrm{BH}_{2i-1}, \mathcal{L}_2 \in \mathrm{coNP}\} = \mathrm{BH}_{2i-1} \bigwedge \mathrm{coNP}$
- $\mathrm{BH}_{2i+1} = \{\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2 \mid \mathcal{L}_1 \in \mathrm{BH}_{2i}, \mathcal{L}_2 \in \mathrm{NP}\} = \mathrm{BH}_{2i} \bigvee \mathrm{NP}$
- $\mathrm{BH} = \cup_{i \in \mathbb{N} \setminus \{0\}} \mathrm{BH}_i \subseteq \Delta_2^{\mathrm{P}}$

Inclusion of Boolean Hierarchy Classes

- $\forall i, \mathrm{BH}_i \subseteq \mathrm{BH}_{i+1}$
- Intuition:
  - For $i \equiv 1 \pmod 2$, $\forall \mathcal{L} \in \mathrm{BH}_i$, $\mathcal{L} = \mathcal{L} \cap \mathcal{L}_{All}$, where $\mathcal{L}_{All}$ is the language which contains all possible instances. $\mathcal{L}_{All} \in$ coNP, so $\mathcal{L} = \mathcal{L} \cap \mathcal{L}_{All} \in \mathrm{BH}_{i+1}$
  - For $i \equiv 0 \pmod 2$, $\forall \mathcal{L} \in \mathrm{BH}_i$, $\mathcal{L} = \mathcal{L} \cup \mathcal{L}_{\emptyset}$, where $\mathcal{L}_{\emptyset}$ is the language which contains no instance at all. $\mathcal{L}_{\emptyset} \in$ NP, so $\mathcal{L} = \mathcal{L} \cup \mathcal{L}_{\emptyset} \in \mathrm{BH}_{i+1}$

A Specific Class from BH: DP

- DP $= \mathrm{BH}_2 = \{\mathcal{L}_1 \cap \mathcal{L}_2 \mid \mathcal{L}_1 \in \mathrm{NP} \text{ and } \mathcal{L}_2 \in \mathrm{coNP}\}$ for Difference Polynomial Time
- DP has been identified before the definition of the Boolean hierarchy [Papadimitriou and Y. 1984]
  - DP $= \{\mathcal{L}_1 \setminus \mathcal{L}_2 \mid \mathcal{L}_1 \in \mathrm{NP} \text{ and } \mathcal{L}_2 \in \mathrm{NP}\}$
- $\mathrm{NP} \subseteq \mathrm{DP}$, $\mathrm{coNP} \subseteq \mathrm{DP}$

A Specific Problem from DP (1)

Exact Clique

Given a graph G and an integer k, is it true that the maximal clique in G has size exactly k?

Complexity of Exact Clique

Exact Clique is DP-complete [Papadimitriou and Y. 1984]

A Specific Problem from DP (2)

SAT-UNSAT

Given two propositional formulas ϕ, ψ, is ϕ satisfiable and ψ unsatisfiable?

Complexity of SAT-UNSAT

SAT - UNSAT is DP-complete [Papadimitriou and Y. 1984]

A Specific Problem from DP (3)

Unique SAT

Given a propositional formula ϕ, is ϕ satisfiable with exactly one model?

Complexity of Unique SAT

Unique SAT is in DP [Papadimitriou and Y. 1984]

References

[Wechsung 1995] G. Wechsung, *On the Boolean closure of* NP. Proc. of the International Conference on Fundamentals of Computation Theory, p 485–493, 1985.

[Papadimitriou and Y. 1984] C. H. Papadimitriou and M. Yannakakis, *The complexity of facets (and some facets of complexity)*. Journal of Computer and System Sciences 28, p 244–259, 1984.

Outline

Intuition of Parameterized Complexity

- When considering a hard problem, sometimes the difficulty to solve it comes from a specific part of the problem.
- If, for an instance i, we are sure that the hard part is not too big, then maybe the instance i is “not so hard”

Intuition of Parameterized Complexity

- When considering a hard problem, sometimes the difficulty to solve it comes from a specific part of the problem.
- If, for an instance *i*, we are sure that the hard part is not too big, then maybe the instance *i* is “not so hard”

Intuitive Example: SAT

- If ϕ is a DNF formula, it is easy to give all its models

Intuition of Parameterized Complexity

- When considering a hard problem, sometimes the difficulty to solve it comes from a specific part of the problem.
- If, for an instance *i*, we are sure that the hard part is not too big, then maybe the instance *i* is “not so hard”

Intuitive Example: SAT

- If ϕ is a DNF formula, it is easy to give all its models
- $\phi \wedge (x_1 \vee x_2 \vee x_3)$ is not a DNF, so it is not (directly) easy to know if it is satisfiable

Intuition of Parameterized Complexity

- When considering a hard problem, sometimes the difficulty to solve it comes from a specific part of the problem.
- If, for an instance i, we are sure that the hard part is not too big, then maybe the instance i is "not so hard"

Intuitive Example: SAT

- If ϕ is a DNF formula, it is easy to give all its models
- $\phi \wedge (x_1 \vee x_2 \vee x_3)$ is not a DNF, so it is not (directly) easy to know if it is satisfiable
- But since we know all the models of ϕ, it is easy to know if one of them satisfies $x_1 \vee x_2 \vee x_3$

Intuition of Parameterized Complexity

- When considering a hard problem, sometimes the difficulty to solve it comes from a specific part of the problem.
- If, for an instance *i*, we are sure that the hard part is not too big, then maybe the instance *i* is "not so hard"

Intuitive Example: SAT

- If ϕ is a DNF formula, it is easy to give all its models
- $\phi \land (x_1 \lor x_2 \lor x_3)$ is not a DNF, so it is not (directly) easy to know if it is satisfiable
- But since we know all the models of ϕ, it is easy to know if one of them satisfies $x_1 \lor x_2 \lor x_3$
- More generally, for $\phi \land \psi$, with ϕ a DNF, the size of ψ is a parameter of the difficulty: the smaller ψ, the easier it is to solve SAT for $\phi \land \psi$

Parameters of a Problem

- Instead of measuring the complexity of a problem with just the "size" of the problem, we consider several parameters
- Each parameter correspond to a different source of difficulty to solve the problem
- If some parameters are fixed (or bounded), then the problem becomes "simpler" than in the general case

Parameterized Language
[Downey and Fellows 1995]

Definition

Given an alphabet Σ, a parameterized language $\mathcal{L}$ is a subset of $\Sigma^* \times \Sigma^*$. For $(x, k) \in \mathcal{L}$, we call x the main part and k the parameter.

Parameterized Language
[Downey and Fellows 1995]

Definition

Given an alphabet Σ, a parameterized language $\mathcal{L}$ is a subset of $\Sigma^* \times \Sigma^*$. For $(x, k) \in \mathcal{L}$, we call x the main part and k the parameter.

Definition

A parameterized language $\mathcal{L}$ is fixed-parameter tractable if it can be decided in $\mathcal{O}(f(k) \times p(n))$, for $(x, k) \in \mathcal{L}$, with $n = |x|$, $p(n)$ a polynomial, and $f(k)$ any function.
The class of fixed-parameter tractable problems is called FPT

Examples of Natural Parameters

- The size of a database query
- The number of variables in a logical formula
- The number of moves in a game [Abrahamson *et al.* 1995]

Examples of Natural Parameters

- The size of a database query
- The number of variables in a logical formula
- The number of moves in a game [Abrahamson *et al.* 1995]
- The number of Boolean variables to be assigned to move a formula in a tractable class

Partial Assignment of Formulas

Partial Assignment

We suppose that V is the set of propositional variables, and $V' \subseteq V$.

- A partial assignment on V' is a mapping ω from each variable $x \in V'$ to $\{0, 1\}$.
- For a propositional formula ϕ, the result of the partial assignment ω on ϕ is ϕ_ω.

Partial Assignment of Formulas

Partial Assignment

We suppose that V is the set of propositional variables, and $V' \subseteq V$.

- A partial assignment on V' is a mapping ω from each variable $x \in V'$ to $\{0, 1\}$.
- For a propositional formula ϕ, the result of the partial assignment ω on ϕ is ϕ_ω.

Example

For $V = \{x_1, x_2, x_3, x_4\}$ and $\phi = (x_1 \vee x_3) \wedge (\neg x_2 \vee x_4)$, we define

- $V' = \{x_1, x_2\}$
- $\omega(x_1) = 0$, $\omega(x_2) = 1$
- $\phi_\omega = ?$

Partial Assignment of Formulas

Partial Assignment

We suppose that V is the set of propositional variables, and $V' \subseteq V$.

- A partial assignment on V' is a mapping ω from each variable $x \in V'$ to $\{0, 1\}$.
- For a propositional formula ϕ, the result of the partial assignment ω on ϕ is ϕ_ω.

Example

For $V = \{x_1, x_2, x_3, x_4\}$ and $\phi = (x_1 \vee x_3) \wedge (\neg x_2 \vee x_4)$, we define

- $V' = \{x_1, x_2\}$
- $\omega(x_1) = 0$, $\omega(x_2) = 1$
- $\phi_\omega = (0 \vee x_3) \wedge (\neg 1 \vee x_4) = x_3 \wedge x_4$

Backdoors in SAT

Target Class

Given $\mathcal{C}$ a class of propositional formulas, $\mathcal{C}$ is a target class if

- $\mathcal{C}$ can be recognized in polynomial time
- satisfiability of formulas in $\mathcal{C}$ can be checked in polynomial time
- $\mathcal{C}$ is closed under isomorphism (*i.e.* is two formulas are identical except for the names of the variables, then either both or none belong to $\mathcal{C}$)

Definition

A strong-$\mathcal{C}$-backdoor of a CNF formula ϕ is a set of variables B such that for all interpretations $\tau \in 2^B$, $\phi_\tau \in \mathcal{C}$

If we know a strong-$\mathcal{C}$-backdoor of ϕ of size k, then the satisfiability of ϕ is reduced to the satisfiability of 2^k formulas $\phi_1, \ldots, \phi_{2^k} \in \mathcal{C}$. So SAT becomes fixed-parameter tractable in k

Examples of Base Classes

Horn Formulas

A Horn formula is a CNF with only Horn clauses, *i.e.* clauses with at most one positive literal

2CNF Formulas

A 2CNF formula is a CNF with only unary and binary clauses

This means that if a formula has a backdoor of size k to Horn or 2CNF, then it is FPT in parameter k

More details in [Gaspers and Szeider 2012]

Example of Backdoor for Horn Formulas

- $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$ is not a Horn formula

Example of Backdoor for Horn Formulas

- $\phi = (x_1 \lor x_2 \lor \neg x_3) \land (x_3 \lor \neg x_4 \lor \neg x_5)$ is not a Horn formula
- For $B = \{x_2\}$, we have two possible partial assignments
  - $\omega_1(x_2) = 0$
  - $\omega_2(x_2) = 1$

Example of Backdoor for Horn Formulas

- $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$ is not a Horn formula
- For $B = \{x_2\}$, we have two possible partial assignments
  - $\omega_1(x_2) = 0$
  - $\omega_2(x_2) = 1$
- $\phi_{\omega_1} = (x_1 \vee \neg x_3) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$ is a Horn formula

Example of Backdoor for Horn Formulas

- $\phi = (x_1 \lor x_2 \lor \neg x_3) \land (x_3 \lor \neg x_4 \lor \neg x_5)$ is not a Horn formula
- For $B = \{x_2\}$, we have two possible partial assignments
  - $\omega_1(x_2) = 0$
  - $\omega_2(x_2) = 1$
- $\phi_{\omega_1} = (x_1 \lor \neg x_3) \land (x_3 \lor \neg x_4 \lor \neg x_5)$ is a Horn formula
- $\phi_{\omega_2} = (x_3 \lor \neg x_4 \lor \neg x_5)$ is a Horn formula

Example of Backdoor for Horn Formulas

- $\phi = (x_1 \lor x_2 \lor \neg x_3) \land (x_3 \lor \neg x_4 \lor \neg x_5)$ is not a Horn formula
- For $B = \{x_2\}$, we have two possible partial assignments
  - $\omega_1(x_2) = 0$
  - $\omega_2(x_2) = 1$
- $\phi_{\omega_1} = (x_1 \lor \neg x_3) \land (x_3 \lor \neg x_4 \lor \neg x_5)$ is a Horn formula
- $\phi_{\omega_2} = (x_3 \lor \neg x_4 \lor \neg x_5)$ is a Horn formula
- So B is a backdoor of size 1 for ϕ and the Horn target class

Graph Treewidth

- Number that (intuitively) indicates how close a graph is to a tree
- Many problems are FPT in the treewidth, *i.e.* they are polynomial if the treewidth of the graph is constant/bounded
- Idea:
  - from a non-directed graph, we can define some *tree decompositions*, that are trees made from the elements of the graph
  - a formula associates a number n to each possible tree decomposition of thre graph
  - the treewidth of the graph is the minimal value of n

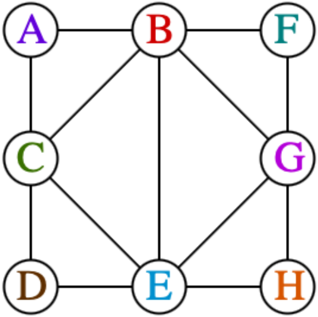
Graph Decomposition

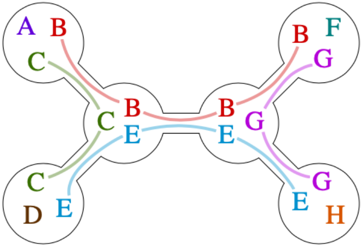
Given a non-directed graph $G = \langle N, E \rangle$, define $D = (T, \lambda)$ where $T = \langle N_T, E_T \rangle$ is a tree, and $\lambda : N_T \to 2^N$ is a mapping from tree nodes to subsets of the graph nodes, such that

- $\forall n \in N$, there is $t \in N_T$ such that $n \in \lambda(t)$ (*i.e.* $\bigcup_{t \in N_T} \lambda(t) = N$).
- $\forall \{x, y\} \in E$, there is $t \in N_T$ such that $\{x, y\} \subseteq \lambda(t)$
- $\forall n \in N$, the nodes $\{t \in N_T \mid n \in \lambda(t)\}$ form a connected subtree

The *width* of a tree decomposition is $w(D) = (\max_{t \in N_T} |\lambda(t)|) - 1$

Example of Graph Decomposition





(source:Wikipedia)

Here, $w(D) = 2$

Treewidth of a Graph

- Every graph has an infinite number of tree decompositions:
  - Trivial decomposition: take any tree T, and $\forall t \in N_T, \lambda(t) = N$
- The treewidth of a graph G is $tw(G) = \min_D w(D)$
- For any graph $G = \langle N, E \rangle$, $1 \leq tw(G) \leq k - 1$ where $k = |n|$
  - $tw(G) = 1$: G is a tree
  - $tw(G) = k - 1$: G is a clique

Treewidth of a Graph

- Every graph has an infinite number of tree decompositions:
  - Trivial decomposition: take any tree T, and $\forall t \in N_T, \lambda(t) = N$
- The treewidth of a graph G is $tw(G) = \min_D w(D)$
- For any graph $G = \langle N, E\rangle$, $1 \leq tw(G) \leq k - 1$ where $k = |n|$
  - $tw(G) = 1$: G is a tree
  - $tw(G) = k - 1$: G is a clique
- Finding the treewidth of a graph is generally hard

Theorem [Arnborg *et al.* 1987]

Given G and $k \in \mathbb{N}$, deciding whether $tw(G) \leq k$ is NP-complete

Treewidth of SAT

- Given any CNF ϕ, we can define a graph $G_\phi = \langle N, E \rangle$ such that N is the set of variables in ϕ, and $\{x, y\} \in E$ iff the variables x and y appear together in some clause
- Define the treewidth of the formula: $tw(\phi) = tw(G_\phi)$

Theorem

SAT is FPT with respect to the treewidth

- More precisely, SAT can be solved in $\mathcal{O}(nk \times 2^k)$ where n is the number of Boolean variables, and $k = tw(\phi)$

Courcelle Theorem

- Monadic Second-Order logic (MSO) is a rich logical framework for reasoning over sets

Theorem [Courcelle 1990]

Any property of a graph expressed in MSO logic is decidable in linear time if the graph has a bounded treewidth.

Graph Coloring, MSO and Treewidth

- k-Coloring Problem: given a non-directed graph $G = \langle N, E \rangle$, and a set $C = \{c_1, \ldots, c_k\}$ of colors, is it possible to color each node of the graph such that neighbours have a different color?
- k-Coloring Problem is NP-complete for any $k \geq 3$
- We can express 3-Coloring Problem in MSO logic:

$$\begin{array}{rl} \exists c_1, c_2, c_3, & (\forall n \in N, (n \in c_1 \vee n \in c_2 \vee n \in c_3)) \\ \wedge(\forall n, n' \in N, & ((n \in c_1 \wedge n' \in c_1) \vee (n \in c_2 \wedge n' \in c_2) \\ & \vee(n \in c_3 \wedge n' \in c_3)) \rightarrow \neg adj(n, n')) \end{array}$$

- So it is FPT with respect to the graph treewidth

References

[Downey and Fellows 1995] R. G. Downey and M. R. Fellows, *Fixed-Parameter Tractability and Completeness I: Basic Results*. SIAM J. Comput. 24(4): 873–921, 1995.

[Abrahamson *et al.* 1995] K. R. Abrahamson, R. G. Downey and M. R. Fellows, *Fixed-Parameter Tractability and Completeness IV: On Completeness for W[P] and PSPACE Analogues*. Ann. Pure Appl. Logic 73(3): 235–276, 1995.

[Gaspers and Szeider 2012] S. Gaspers and S. Szeider, *Backdoors to Satisfaction*. The Multivariate Algorithmic Revolution and Beyond, 287–317, 2012.

References

[Arnborg *et al.* 1987] S. Arnborg, D. Corneil and A. Proskurowski, *Complexity of Finding Embeddings in a k-Tree*. SIAM Journal on Algebraic Discrete Methods. 8(2): 277–284, 1987.

[Courcelle 1990] B. Courcelle, *The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs*. Information and Computation. 85(1): 12–75, 1990.