Fixed Parameter Tractability of SAT through Backdoors

Rupert Ettrich
Matr.: 01129393
Curriculum: 066 931 Logic and Computation
rupert.ettrich@gmail.com

June 14, 2021

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Write abstract

1 Introduction

The Satisfiability Problem (SAT) is a fundamental problem in logic and computer science. Given a formula F in propositional logic, the task is to determine whether there exists some assignment of variables in F to the truth values $\{0, 1\}$ such that the formula is evaluated to 1. The simplicity of its definition makes SAT a very powerful modeling tool for many different problems related to software and hardware planning and design. For many such problems it is easier to express it as a propositional logic formula where a satisfying variable assignment can be transformed into a solution of the original problem, and use a SAT-solver on the instance than to develop a problem-specific algorithm.

The famous Cook-Levin Theorem [] states that SAT belongs to the class of NP-Complete problems. This means that, although it is easy to verify a solution, the existence of an efficient polynomial time algorithm to solve any given instance seems highly unlikely. However, despite the theoretical worst-case complexity of the problem, many large real-world SAT instances can be solved rather efficiently by modern SAT-solvers. This gap between theoretical and practical results lead to research on the structural properties of SAT-instances that can be exploited in order to explain this phenomenon.

The framework of Parameterized Complexity was introduced by Downey and Fellows [] in order provide tools for a more fine-grained analysis of computationally hard problems. This framework allows to consider the runtime of a problem given some fixed parameter. Depending on the chosen parameter, the runtime can differ drastly, which leads to the study of fixed-parameter tractable parameters. In the context of SAT this leads to two (not necessarily exclusive) approaches when studying its parameterized complexity: Structural Decomposition and Backdoors. While the former is comprised of approaches to find and exploit structural properties of the whole input formula, the latter is about finding a small subset of variables whose assignment leads into a tractable class of SAT-instances. In this seminar paper we will focus on showing fixed-parameter tractability results that can be achieved by the latter approach, Backdoors.

Section 2 contains information about the necessary preliminaries and notation that is used in this work. Section 3 gives a general overview of fixed-parameter tractability and intractability results regarding Backdoors in SAT and is based on a survey of Gaspers and Szeider [1]. Section 4 contains recent contributions to research regarding Backdoors in SAT. Finally, section 5 summarizes the results shown in this seminar paper and offers some concluding remarks.

2 Preliminaries

This section contains definitions of the general concepts used in context with Backdoors in SAT. We conform to the notation used by Gaspers and Szeider [1]. Section 4 contains additional definitions and notation specific to the presented material.

2.1 The Satisfiability Problem

The Satisfiability Problem, Boolean Satisfiability Problem, or SAT is a fundamental problem in computer science with many applications in software and hardware planning and scheduling, since many problems can be modeled as SAT-instances and solved by SAT-solvers. It is an NP-Complete decision problem that can be defined as follows:

SAT
Input: A propositional logic formula F in conjunctive normal form (CNF) over propositional variables $X = \{x_1, x_2, ..., x_n\}$
Question: Is there a truth assignment $\tau : X \rightarrow \{0,1\}$ (or $\tau \in 2^X$) such that $F[\tau]$ evaluates to 1?

We use the following notation: A literal is a propositional variable x or x^1 (positive literal) or a negated variable $\overline{x}$ or x^0 (negative literal). A clause is a finite set of literals. A SAT-instance is comprised of a propositional logic formula F in conjunctive normal form (CNF), which is a set of clauses, where the literals inside each clause are only connected by the binary operator "logical or" ($\vee$) and the clauses are connected only by the binary operator "logical and" ($\wedge$). We denote as $k - CNF$ the set of propositional logic formulae in conjunctive normal form that contain at most k literals in each clause. When describing a formula F, we use the following equivalent notations: $F = \{\{x_1, \overline{x_2}\}, \{x_3\}\} = (x_1 \vee \overline{x_2}) \wedge x_3$. We call a clause C containing only positive literals a *postivie clause*, and a clause containing only negative literals a *negative clause*.

If not stated otherwise, we use F to describe a formula, and $C \in F$ to describe a clause. We denote as $var(F)$ the set of all variables that appear in a formula F as a positive or negative literal. Similarly, we denote as $var(C)$ the set of variables that appear in a clause C as a positive or negative literal. A truth assignment $\tau : X \rightarrow \{0,1\}$ is a function that maps the variables in $X \subseteq var(F)$ to the truth values $\{0,1\}$ and we denote as 2^X the set of all truth assignments over X. For $\tau \in 2^X$ let $\mathrm{true}(\tau) = \{x^{\tau(x)} : x \in X\}$ and $\mathrm{false}(\tau) = \{x^{1-\tau(x)} : x \in X\}$ be the set of of literals set by τ to 1 and 0 respectively. We then define $F[\tau] = \{C \setminus \mathrm{false}(\tau) : C \in F, C \cap \mathrm{true}(\tau) = \emptyset\}$. Informally, $F[\tau]$ is the set of clauses that remain after we removing all clauses containing at least one literal that is evaluated to 1 by τ, and removing all literals that are evaluated to 0 by τ from these remaining clauses. A CNF-formula F is *satisfiable* if there is some $\tau \in 2^{var(F)}$ such that $F[\tau] = \emptyset$, so each clause $C \in F$ contains at least one literal that evaluates to 1 under τ. If there exists no such $\tau \in 2^var(F)$, the formula is *unsatisfiable*. SAT then becomes the problem of deciding whether a CNF-formula F is satisfiable.

We conclude these definitions with a small example. Let $F = (x_1 \vee x_2) \wedge (\overline{x_3} \vee x_5) \wedge \overline{x_5}$, $X_1 = \{x_1, x_5\}$, $\tau_1 \in 2^{X_1}$ s.t. $\tau_1[x_1] = 0, \tau_1[x_5] = 1$. Then $F[\tau_1] = \{\{x_2\}, \{\}\}$ and thus $F[\tau_1]$ is unsatisfiable, as it contains an empty clause. However, for $X_2 = \{x_1, x_3, x_5\}, \tau_2 \in X_2$ s.t. $\tau_2[x_1] = 1, \tau_2[x_3] = 0, \tau[x_5] = 0$ it can easily be seen that $F[\tau_2] = \emptyset$, as each clause is satisfied and therefore F is satisfiable and τ_2 is a satisfying variable assignment.

add primal, hypergraph of F

2.2 Fixed Parameter Tractability

The framework of Parameterized Complexity was introduced by Downey and Fellows [] . In this section we list the definitions of a parameterized problem, fixed-parameter tractability, and W-hardness as defined by Cygan et al. [].

cite

Definition 2.1 ([]). A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet. For an instance $(x, k) \in \Sigma^*$, k is called the *parameter*. cite

In the context of SAT, this definition corresponds to a pair (F, k), which is a CNF-formula F and some parameter k, that is a positive integer that denotes some structural property of the formula, e.g. the treewidth of the primal graph of F.

Definition 2.2 ([]). A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is called *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$ (called a *fixed-parameter algorithm*), a computable function $f : \mathbb{N} \to \mathbb{N}$, and a constant c such that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^c$. The complexity class containing all fixed-parameter tractable problems is called FPT. cite

Informally, fixed-parameter tractability can be used as a tool to investigate the complexity of NP-Hard problems in more detail. If an NP-Complete problem is FPT, it means that the combinatorial explosion in the runtime that is expected by NP-Complete problems is restricted to the parameter k. Therefore, by fixing the parameter k to a constant value, we obtain a runtime that is polynomial in the size of the input. For SAT, there are many such parameters, e.g. the treewidth of the primal graph of a CNF-formula F, or the size k of a strong backdoor set into a tractable class of formulas. However determining the value of the parameter k might be a NP-Complete problem itself (like determining the treewidth of a formula F). Furthermore, the selection of the parameter k matters, as some parameters lead to FPT-results, while others do not. This is because there are problems which are assumed to not be FPT. The theory of W-hardness can be utilized to show that a problem is most likely not FPT [] . cite This is done by finding a parameterized reduction, that is, a reduction in FPT time, from a W-hard problem to the problem for which fixed-parameter intractability is to be shown.

2.3 Base Classes

2.4 Backdoor Sets

3 General Results

The term "Backdoor" was coined by Williams et al. [2]

3.1 Strong Backdoor Set Detection

3.2 Weak Backdoor Set Detection

4 Related Work

5 Conclusions

References

[1] Serge Gaspers and Stefan Szeider. *Backdoors to Satisfaction*, pages 287–317. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[2] Richard Williams, Carla Gomes, and Bart Selman. Backdoors To Typical Case Complexity. *IJCAI International Joint Conference on Artificial Intelligence*, 09 2003.