Chapter 1

Introduction

Parameterized complexity \[26\] works by trying to find efficient algorithms for instances of hard problems where one can identify structure that helps in analysis. A numerical parameter (usually denoted by \(k\)) is associated with problem instances and algorithms are designed whose time and/or memory requirement is a fast growing function of the parameter, but growing slowly in terms of the size of the instance. On instances where the parameter is small, such algorithms run efficiently. For example, with hard problems for which all known algorithms have worst case running time exponential in the size \(n\) of the input size, parameterized complexity looks for algorithms with worst case running time \(f(k)n^c\), where \(f\) is some computable function of the parameter \(k\) and \(c\) is a constant. Parameterized problems with algorithms of such running time are in the class Fixed Parameter Tractable (Fpt), which can be thought of as parameterization of the classical complexity class Polynomial time (Ptime). Appendix A contains a brief introduction to concepts related to Fpt and a hierarchy of parameterized complexity classes believed to be intractable. Other classical complexity classes can also be parameterized \[36\]. There are articles in the literature that consider questions that are essentially parameterized complexity problems related to concurrent systems \[87, 45\]. We refine such results by considering other parameters and utilizing techniques that have been recently developed in the field of parameterized complexity.

The study of parameterized complexity derived an initial motivation from the study of graph parameters. Many problems that are complete for non-deterministic polynomial time (NP) can be solved in polynomial time on trees and are Fpt on graphs that have tree-structured decompositions.

Definition 1.1 (Tree decomposition, treewidth, pathwidth). A tree decomposition of a graph \(G = (V,E)\) is a pair \((T, (B_t)_{t \in \text{nodes}(T)})\), where \(T\) is a tree and \((B_t)_{t \in \text{nodes}(T)}\) is a family of subsets of \(V\) such that:

- For all \(v \in V\), the set \(\{t \in \text{nodes}(T) \mid v \in B_t\}\) is nonempty and connected in \(T\).
- For every edge \((v_1,v_2) \in E\), there is a \(t \in \text{nodes}(T)\) such that \(v_1, v_2 \in B_t\).

The width of such a decomposition is the number \(\max\{|B_t| \mid t \in \text{nodes}(T)\} - 1\). The treewidth \(tw(G)\) of \(G\) is the minimum of the widths of all tree decompositions of \(G\). If the tree \(T\) in the definition of tree decomposition is a path, we get a path decomposition. The pathwidth \(pw(G)\) of \(G\) is the minimum of the widths of all path decompositions of \(G\).

Treewidth is a well-studied graph parameter that arises naturally in many contexts. For example, Thorup has shown that flow graphs resulting from structured programs of many languages have small treewidth \[93\]. Treewidth has also been used as a unifying framework for many decidability results for automata with auxiliary storage \[67\] and for efficient model checking of First Order logic \[2\]. Treewidth being such a fundamental and widely occurring
concept, we study its impact on the satisfiability of modal and temporal logics. Modal logics have many applications (reasoning about knowledge [31], programming [81] and hardware verification [84] etc.), in addition to nice computational properties [97, 41]. Many tools have been built for checking satisfiability of modal formulas [33, 77], despite being intractable in the classical sense (PSPACE-complete or NP-complete in most cases). Complexity of modal logic decision problems is well studied [60]. Temporal logic is widely used for formal specification and verification of concurrent systems and extensively researched [15, 98, 68, 59].

In the first part of this thesis, we study the effect of treewidth as a parameter for the satisfiability problems of modal logic, Linear Temporal Logic (LTL) and Computational Tree Logic (CTL). The LTL satisfiability problem is known to be polynomial space (PSPACE)-complete [90] and CTL satisfiability problem is known to be exponential time (EXPTIME)-complete [28]. For modal logic, the satisfiability problem is usually PSPACE-complete or non-deterministic polynomial time (NP)-complete [60], depending on the restrictions imposed on satisfying models. We look for FPT algorithms or for hardness for some parameterized complexity class believed to be intractable (details in Appendix A). It turns out that when there is transitivity or some equivalent concept such as in LTL, CTL and modal satisfiability in transitive models, treewidth does not help. We will see that treewidth does help in the general modal satisfiability problem, thus gathering some evidence that it is transitivity that makes treewidth useless, as far as FPT algorithms are concerned.

Labelled Transition Systems (LTS) are models of sequential systems while extensions like Synchronized LTS and 1-safe Petri nets compactly represent concurrent finite state systems. We continue to study parameterized complexity of problems associated with these models. Most of the problems we consider are PSPACE-complete.

Petri nets, introduced by C. A. Petri [80], are popularly used for modelling concurrent infinite state systems. Using Petri nets to verify various properties of concurrent systems is an ongoing area of research, with abstract theoretical results like [5] and actually constructing tools for C programs like [50]. Reachability is one of the most fundamental problems of Petri nets. Although it is known to be decidable [71, 58], the complexity is not known. Complexity of the reachability problem is known for many subclasses of Petri nets which are the result of various restrictions on Petri nets. Finding such restrictions helps in understanding the structure of Petri nets and in developing techniques. We use parameterized complexity as a mathematically rigorous way of analyzing structural restrictions. With the capability of having an “extended dialog” with hard problems [25], answers to parameterized complexity questions can provide finer understanding of the problems under consideration.

Apart from reachability, coverability and boundedness are some of the most fundamental questions about Petri nets. All three of them are exponential space (EXPSPACE)-hard [60]. Coverability and boundedness are in EXPSPACE [55]. Reachability is known to be decidable [71, 58, 62, 65] but no upper bound is known. Other interesting properties of Petri nets include liveness, deadlock, fairness etc., which especially arise in the context of concurrent reactive systems. Logics have been proposed to uniformly describe such properties of Petri nets [52, 77, 14, 93]. In the second part of this thesis, we study the parameterized complexity of the coverability and boundedness problems. We also study the parameterized complexity of model checking a logic that we have carefully designed to avoid expressing the reachability problem, but is powerful enough to express coverability, boundedness and some extensions.

Let us denote the size of an input instance of a problem by $n$. If the problem is EXPSPACE-hard, any algorithm will require memory space exponential in $n$ in the worst case. Let us consider some parameter denoted by $k$. For an EXPSPACE-hard problem such as those mentioned in the previous paragraph, a suitable parameterized complexity theoretic question would be to check if there are algorithms solving the problem using memory space

\footnote{A survey can be found in [82]}
The function $f(k)$ may be any computable function of the parameter $k$ while $\text{poly}(n)$ is some polynomial of the input size. Such algorithms are called \textsc{Paraspac}e algorithms. Fundamental complexity theory of such parameterized complexity classes have been studied \cite{36}. We consider two parameters and provide \textsc{Paraspac}e algorithms for coverability, boundedness and model checking the newly designed logic.

This thesis is organized as follows. In Chapter 2, we study the extent to which treewidth can help in obtaining Fixed Parameter Tractable (FPT) algorithms for satisfiability of various modal logics. In Chapter 3, we give some parameterized complexity results for synchronized transition systems. In Chapter 4, we consider 1-safe Petri nets. Here, the problems under consideration are PSPACE-complete and we look for FPT algorithms (or their absence). We give a brief survey of the literature on concepts and models related to Petri nets and their logics in Chapter 5. In Chapter 6, we introduce a parameter that we call benefit depth, motivated by some refinements in Rackoff’s EXPSPACE upper bound \cite{83} for coverability and boundedness. In Chapter 7, we consider another parameter vertex cover and its effect on the complexity of coverability and boundedness. We summarize our results in Chapter 8.

Appendix A gives a brief introduction to parameterized complexity and some intuition about why parameter treewidth makes certain problems easier.

Some of the work mentioned above have been published in the following papers.


