The Parameterized Complexity of Some Problems in Logic and Linguistics

Extended Abstract – March 1, 2002

Rodney G. Downey
Department of Mathematics, Victoria University
P.O. Box 600, Wellington, New Zealand
downey@math.vuw.ac.nz

Michael R. Fellows, Bruce M. Kapron and Michael T. Hallett
Department of Computer Science, University of Victoria
Victoria, British Columbia V8W 3P6 Canada
contact author: mfellows@csr.uvic.ca

H. Todd Wareham
Department of Computer Science
Memorial University of Newfoundland
St. Johns, Newfoundland A1C 5S7 Canada
harold@odie.cs.mun.ca

Summary

The theory of parameterized computational complexity introduced in [DF1-3] appears to be of wide applicability in the study of the complexity of concrete problems [ADF,BFH,DEF,FHW,FK]. We believe the theory may be of particular importance to practical applications of logic formalisms in programming language design and in system specification. The reason for this relevance is that while many computational problems in logic are extremely intractable generally, realistic applications often involve a "hidden parameter" according to which the computational problem may be feasible according to the more sensitive criteria of *fixed-parameter tractability* that is the central issue in parameterized computational complexity. We illustrate how this theory may apply to problems in logic, programming languages and linguistics by describing some examples of both tractability and intractability results in these areas. It is our strong expectation that these results are just the tip of the iceberg of interesting applications of parameterized complexity theory to logic and linguistics. The main results described in this abstract are as follows. (1) The problem of determining whether a word x can be derived in k steps in a context-sensitive grammar G (*Short CSL Derivation*) is complete for the parameterized complexity class $W[1]$. (2) The problem *Minimum Axiom Set* [GJ] is complete for the parameterized complexity class $W[P]$. The latter result has the added significance of providing a starting point for a proof of the important structural result $W[P] =$ monotone $W[P]$. We also report on the parameterized complexity of reachability in Petri nets (equivalently, vector addition systems) and of some problems concerning phonological derivation in linguistic theory.

1. Introduction

We may begin by making the following general observations concerning computational problems.

(1) Many important computational problems have input consisting of two or more parts. Familiar combinatorial examples include *Bandwidth*, *Min Cut Linear Arrangement*, *Vertex Cover*, *Clique* and *Dominating Set* [GJ], all of which take as input a graph G and a positive integer k. *Minor Testing* takes as input a "host" graph G and a "guest" graph H. An example in logic is *Model Checking* , which takes as input a formula ϕ and a structure M, and determines if M is a model for ϕ. For problems such as these it is natural to consider the relative contribution of the different parts of the input to the complexity of the problem.

(2) There is a large and growing body of results of the following qualitative sort: there is a constant α such for every fixed "parameter" k, the problem Π is solvable in time $O(n^{\alpha})$, with α independent of k. For example, *Min Cut Linear Arrangement* and *Vertex Cover* are solvable in time linear in the number of vertices of G for every fixed k. *Minor Testing* can be solved in time approximately $500^{|H|}|G|^3$, that is, in time cubic in the size of G for every fixed "parameter" H; the techniques used to prove such *fixed parameter tractability* results are in some cases both deep and and distinctive. A nice example of fixed-parameter tractability in logic is given by Lichtenstein and Pnueli [LP], who show that model checking in PLTL (Propositional Linear Temporal Logic) can be done in time $O(2^{|\phi|}|M|)$, that is, in time linear in the size of the structure M for each fixed formula ("system specification") ϕ.

(3) The fact that a problem Π taking as input two items is complete for *NP* (or *PSPACE*, or *EXPTIME*, ...) tells us *nothing* about the fixed-parameter tractability or intractability of Π.

(4) For many natural problems, a small range of parameter values may serve important applications. For example, it has been proposed that the syntactic structure of sentences of natural language can be modeled by dependency graphs of pathwidth no more than 6 (corresponding in some sense to the "bandwidth" of human attention, and modeled computationally by a bounded-depth stack) [KT].

(5) Many natural problems involve a "hidden parameter." Elucidating the contribution of this parameter to the complexity of the problem can sometimes illuminate why the problem is "easier to solve in practice" than one might expect from traditional complexity analysis. An example of this phenomena is the complexity of *Type Inference in ML* which was shown to be complete for $EXPTIME$ in [HM]. It can be shown that for every fixed bound k on the nesting depth of *let*'s in ML formulae, type inference can be accomplished in linear time, providing an explanation for the discrepancy between the very pessimistic completeness result and the behaviour encountered in practice. Engineering considerations may also sometimes introduce (and fix) a parameter. This can render the complexity of a problem *easier* than one desires (for example, in cryptography [FK]). The familiar practice of "coping with NP-completeness" by restricting the input can often be conveniently described by a

parameterization of the input to the problem.

(6) For many familiar parameterized problems the difference between fixed-parameter tractability and apparent intractability is very much akin to the apparent difference we observe between problems which are in P and problems which are NP-complete. For example, while we can determine whether a graph has a vertex cover of size k or a cutwidth k layout in linear time for each fixed k, the best known algorithms for determining whether a graph has a dominating set of size k, or a bandwidth k layout, require time $O(n^{k+1})$ and are based essentially on a brute force examination of all possible solutions. This is strongly reminiscent of the current situation regarding the best known algorithms for many NP-complete problems.

Our main results concern the following problems.

Minimum Axiom Set
Input: A finite set S of "sentences," an "implication relation" R consisting of pairs (A, t) where $A \subseteq S$ and $t \in S$, and a positive integer k.
Parameter: k
Question: Is there a set $S_0 \subseteq S$ with $|S_0| \leq k$ and a positive integer n such that if we define S_i, $1 \leq i \leq n$, to consist of exactly those $t \in S$ for which either $t \in S_{i-1}$ or there exists a set $U \subseteq S_{i-1}$ such that $(U, t) \in R$, then $S_n = S$?

Short CSL Derivation
Input: A context-sensitive grammar $G = (N, \Sigma, \Pi, S)$, a word $x \in \Sigma^*$ and an integer k.
Parameter: k
Question: Is there a G-derivation of x of length at most k?

k-Step Petri Net Reachability
(Defined in §3.)

The length of this abstract does not allow for details of the proofs; the details are intricate combinatorial reductions of the kinds described in [DEF,DF1,DF2,DF4,FHW]. In the §2 we quickly survey the framework of parameterized computational complexity and what is presently known concerning the W hierarchy. In §3 we state the main results that are proved in the paper this abstract represents. In §4 we offer some closing remarks.

2. Parameterized Computational Complexity

The formal framework for our study is established as follows.

Definition. A *parameterized problem* is a set $L \subseteq \Sigma^* \times \Sigma^*$ where Σ is a fixed alphabet.

In the interests of readability, and with no effect on the theory, we consider that a parameterized problem L is a subset of $L \subseteq \Sigma^* \times N$. For a parameterized problem L and $k \in N$ we write L_k to denote the associated fixed-parameter problem (k is the parameter) $L_k = \{x | (x, k) \in L\}$.

Definition. We say that a parameterized problem L is (uniformly) *fixed-parameter tractable* if there is a constant α and an algorithm Φ such that Φ decides if $(x, k) \in L$ in time $f(k)|x|^{\alpha}$ where $f : N \to N$ is an arbitrary function.

Definition. Let A, B be parameterized problems. We say that A is (uniformly many:1) *reducible* to B if there is an algorithm Φ which transforms (x, k) into $(x', g(k))$ in time $f(k)|x|^{\alpha}$ (a *parameterized polynomial-time transformation*), where $f, g : N \to N$ are arbitrary functions and α is a constant independent of k, so that $(x, k) \in A$ if and only if $(x', g(k)) \in B$.

It is easy to see that if A reduces to B and B is fixed parameter tractable then so too is A. Note that if $P = NP$ then problems such as Minimum Dominating Set are fixed-parameter tractable. Thus a completeness program is reasonable.

The classes of parameterized problems that we define below are intuitively based on the complexity of the circuits required to check a solution, or alternatively the "natural logical depth" of the problem. (See also [CC1-2], [KT] and [PY] for differing but complementary views of this issue in terms of alternating logarithmically bounded Turing Machines and finite model theory.)

We first define circuits in which some gates have bounded fan-in and some have unrestricted fan-in. It is assumed that fan-out is never restricted.

Definition. A Boolean circuit is of *mixed type* if it consists of circuits having gates of the following kinds.
(1) *Small gates*: *not* gates, *and* gates and *or* gates with bounded fan-in. We will usually assume that the bound on fan-in is 2 for *and* gates and *or* gates, and 1 for *not* gates.
(2) *Large gates*: *And* gates and *Or* gates with unrestricted fan-in.

Definition. The *depth* of a circuit C is defined to be the maximum number of gates (small or large) on an input-output path in C. The *weft* of a circuit C is the maximum number of large gates on an input-output path in C.

Definition. We say that a family of decision circuits F has *bounded depth* if there is a constant h such that every circuit in the family F has depth at most h. We say that F has *bounded weft* if there is constant t such that every circuit in the family F has weft at most t. The *weight* of a boolean vector x is the number of 1's in the vector.

Definition. Let F be a family of decision circuits. We allow that F may have many different circuits with a given number of inputs. To F we associate the parameterized circuit problem $L_F = \{(C, k) : C \text{ accepts an input vector of weight } k\}$.

Definition. A parameterized problem L belongs to $W[t]$ if L reduces to the parameterized circuit problem $L_{F(t,h)}$ for the family $F(t, h)$ of mixed type decision circuits of weft at most t, and depth at most h, for some constant h.

Definition. A parameterized problem L belongs to $W[P]$ if L reduces to the circuit problem L_F, where F is the set of all circuits (no restrictions). A parameterized problem L belongs to monotone $W[P]$ if L reduces to the circuit problem $L_{F_{mon}}$, where F_{mon} is the set of all monotone circuits.

Definition. We designate the class of fixed-parameter tractable problems FPT.

The above leads to an interesting hierarchy of parameterized complexity classes

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq \text{ mon } W[P] \subseteq W[P]$$

for which a wide variety of natural problems are now known to be complete or hard for various levels (compendiums can be found in [DF2,4,7]). One of the (serendipitous) results of this paper is a proof that monotone $W[P] = W[P]$.

It should be emphasized that there is no easy correspondence between membership or hardness in any of these parameterized complexity classes and NP- or *PSPACE*-completeness (etc.) for the corresponding "unparameterized" problems. For example, in [ADF] problems concerning k-move games are studied, all of which are *PSPACE*-complete in unparameterized form; some of these turn out to be fixed parameter tractable, and others to be complete for $W[P]$. A contrasting example can be found in [DEF] where it is shown that *Vapnik-Chervonenkis Dimension* is complete for $W[1]$; the unparameterized form of this problem is probably of difficulty intermediate between P and NP (see [PY]). Similarly, *Tournament Dominating Set*, which can be solved in time $O(n^{\log n})$ and is unlikely to be NP-complete [PY], can be shown to be complete (in the natural parameterized form) for $W[2]$ (see [DF6]).

The presently known concrete completeness and hardness results are *partially* summarized in the following table.

hard	*complete*	*class*
k-space NDTM Halting		$AW[P]$
	Generalized Geography (k-move winning strategy)	$AW[*]$
	Minimum Axiom Set	$W[P]$
Bandwidth Longest Common Subsequence (for k sequences) DNA Physical Mapping Perfect Phylogeny		$W[t]\ \forall t$
Short Factorization in Monoids k-Processor Scheduling	Minimum Dominating Set	$W[2]$
k-Subset Sum k-Subset Product Perfect Codes in Graphs Short Factorization in Permutation Groups k-Step Petri Net Reachability	k-time NDTM Halting Independent Set Square Tiling Short Post Correspondence Problem k-Step Generation for Context Sensitive Grammars	$W[1]$

3. Main Results

Space limitations only allow the theorems to be stated. The proofs are relatively intricate combinatorial reductions of the kind described in [FHW]. In general, the theory of parameterized computational complexity makes greater technical demands than corresponding investigations in theory of NP-completeness. Nowhere is this more evident than in the only partly resolved question of a parameterized analog of Ladner's density theorem [DF5]. Many of the completeness and hardness results for the W hierarchy that are presently known are the result of several rounds of gradual improvement involving ever more complex combinatorial encoding (the Perfect Phylogeny problem is a good example). The connections between W hardness results for problems for which a fixed parameter value implies bounded combinatorial width (e.g., hypergraph treewidth or pathwidth) are explored in [BFW,FHW], and it is shown that guidance in finding reductions can sometimes be obtained by first proving intractability results with respect to a bounded width hypergraph version of the Myhill-Nerode theorem of formal languages [FHW].

Theorem 1. *Minimum Axiom Set* is complete for $W[P]$.

Corollary $W[P] =$ monotone $W[P]$.

Recall that a grammar $G = (N, \Sigma, \Pi, S)$ is *context-sensitive* if each production rule $(\alpha \rightarrow \beta) \in \Pi$ satisfies $|\beta| \geq |\alpha|$.

Theorem 2. *Short CSL Derivation* is complete for $W[1]$.

We also consider the problem of k-step reachability in Petri nets (or isomorphically, vector addition systems) [GJ]. The problem is defined as follows.

Reachability Distance for Vector Addition Systems (Petri Nets)
Input: A set T of m length n integer-valued vectors $T = \{x^i = (x^i_1, ..., x^i_n) : 1 \leq i \leq m\}$, a non-negative starting vector $s = (s_1, ..., s_n)$, a non-negative target vector $t = (t_1, ..., t_n)$ and a positive integer k.
Parameter: k
Question: Is there a choice of k indices $i_1, ..., i_k$, $1 \leq i_j \leq m$ for $j = 1, ..., k$ such that

$$t = s + \sum_{j=1}^{k} x^{i_j}$$

and such that every intermediate sum is non-negative in each component, that is,

$$s_r + \sum_{j=1}^{q} x^{i_j}_r \geq 0$$

for $q = 1, ..., k$ and $r = 1, ..., n$?

Theorem 3. *Reachability Distance for Vector Addition Systems and Petri Nets* is hard for $W[1]$.

We show that a number of problems in phonological derivation are hard for various levels of the W hierarchy. The relevant models of phonological production are described in [Ris93a,b].

6. Concluding Remarks

In this paper we describe a collection of initial results in the application of parameterized complexity theory to problems in logic and linguistics. The value of this at present is to indicate the kinds of issues the theory can potentially address.

The structural result $W[P] =$ monotone $W[P]$ that is a serendipitous consequence of this initial exploration, is a nice example of the relatively strong role that particular combinatorial reductions have played in elucidating the structure of the W hierarchy.

References

[ADF] K. Abrahamson, R. Downey and M. Fellows, "Fixed-Parameter Intractability II," *Proc. 10th Symposium on Theoretical Aspects of Computer Science (STACS)* (1993), 374–385, Springer-Verlag, Berlin, Lecture Notes in Computer Science.

[BFH] H. Bodlaender, M. Fellows and M. Hallett, "Beyond NP-Completeness for Problems of Bounded Width: Hardness for the W Hierarchy," to appear.

[CC1] L. Cai and J. Chen, "On the Amount of Nondeterminism and the Power of Verifying," to appear in *Proc. International Conference on the Mathematical Foundations of Computer Science (MFCS)*, 1993.

[CC2] L. Cai and J. Chen, "On Fixed-Parameter Tractability and Approximability of NP-hard Optimization Problems," to appear in *Proc. Israeli Conf. on Theoretical Computer Science (ISTCS)*, 1993.

[DEF] R. G. Downey, P. A. Evans and M. R. Fellows, "Parameterized Learning Complexity," *Proc. Sixth ACM Workshop on Computational Learning Theory (COLT)* (1993), 51–57.

[DF1] R. G. Downey and M. R. Fellows, "Fixed Parameter Tractability and Completeness," *Congr. Num.*, 87 (1992) 161-187.

[DF2] R. G. Downey and M. R. Fellows, "Fixed Parameter Tractability and Completeness I: Basic Results," to appear in *SIAM J. Comp.*

[DF3] R. G. Downey and M. R. Fellows, "Fixed Parameter Tractability and Completeness II: On Completeness for $W[1]$," to appear in *TCS-A*.

[DF4] R. G. Downey and M. R. Fellows, "Fixed Parameter Intractability (Extended Abstract)," *Proceedings of the Seventh Annual IEEE Conference on Structure in Complexity Theory* (1992), 36-49.

[DF5] R. G. Downey and M. R. Fellows, "Fixed Parameter Tractability and Completeness III:

Some Structural Aspects of the W-Hierarchy," to appear in *Proc. 1992 Dagstuhl Workshop on Structural Complexity Theory* (Cambridge University Press).

[DF6] R. G. Downey and M. R. Fellows, "Parameterized Computational Feasibility," to appear in *Proc. Second Cornell Workshop on Feasible Mathematics* (Birkhauser, Boston).

[DF7] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, monograph in preparation.

[FHW] M. R. Fellows, M. T. Hallett and H. T. Wareham, "DNA Physical Mapping: Three Ways Difficult," *Proc. First European Symposium on Algorithms* (1993), 157–168.

[FK] M. R. Fellows and N. Koblitz, "Fixed-Parameter Complexity and Cryptography," *Proceedings of the Tenth International Conference on Algebraic Algorithms and Error-Correcting Codes (AAECC 10)*, Springer-Verlag, Lecture Notes in Computer Science, 1993.

[GJ] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).

[HM] F. Henglein and H. G. Mairson, "The Complexity of Type Inference for Higher-Order Typed Lambda Calculi." In *Proc. Symp. on Principles of Programming Languages (POPL)* (1991), 119-130.

[KT] A. Kornai and Z. Tuza, "Narrowness, Pathwidth and Their Application in Natural Language Processing," *Discrete Applied Mathematics* 36 (1992), 87-92.

[KoTh] P. G. Kolaitis and M. N. Thakur, "Approximation Properties of NP Minimization Classes," *Proc. 6th Structure in Complexity Theory Conference* (1991), 353-366.

[LP] O. Lichtenstein and A. Pneuli, "Checking that Finite-State Concurrent Programs Satisfy Their Linear Specification," in *Proc. 12th Ann. ACM Symp. on Principles of Programming Languages* (1985), 97–107.

[PY] C. H. Papadimitriou and M. Yannakakis, "On the Complexity of Computing the V-C Dimension," *Proccedings of the 1993 IEEE Conf. on Structure in Complexity Theory*, (1993).

[Ris93a] E. Ristad, *The Language Complexity Game*, MIT Press, Cambridge, MA, 1993.

[Ris93b] E. Ristad, "Complexity of the Simplified Segmental Phonology", CS-TR-388-92, revised May 1993. Submitted to *Computational Linguistics*.