The Resolution Width Problem is EXPTIME-Complete

Alexander Hertel & Alasdair Urquhart *

November 24, 2007

Abstract

The importance of *width* as a resource in resolution theorem proving has been emphasized in the work of Ben-Sasson and Wigderson. Their results show that lower bounds on the size of resolution refutations can be proved in a uniform manner by demonstrating lower bounds on the width of refutations, and that there is a simple dynamic programming procedure for automated theorem proving based on the search for small width proofs. In addition, if a practical algorithm for calculating width existed, then researchers working with SAT-solvers would be able to use it to predict whether or not to even attempt solving a given formula. Therefore there are both theoretical as well as practical motivations for better understanding resolution width.

Given a set of clauses Σ and an integer k, the resolution width problem is that of determining whether Σ has a refutation of width k. This problem was conjectured to be $\mathcal{EXPTIME}$-complete by Moshe Vardi; the present paper confirms Vardi's conjecture. The proof is by a reduction from the $(\exists, k)$-pebble game, which was shown to be $\mathcal{EXPTIME}$-complete by Kolaitis and Panttaja.

1 Introduction & Motivation

The *width* of a resolution proof is the maximum width of any clause occurring in it. The width of proofs has played a key part in investigations of the complexity of resolution, starting with Tseitin [9] and Galil [5]. In these early papers, and again in the seminal paper by Haken [6], lower bounds on the size of resolution proofs rest on lower bounds on width.

This connection between size and width was systematized and explained in a remarkable paper by Ben-Sasson and Wigderson [3]. In that article they prove a general theorem, of which one consequence is that if Σ is a contradictory set of clauses in 3-CNF, then the minimal size of a resolution refutation of Σ is

*The authors gratefully acknowledge NSERC and the University of Toronto Department of Computer Science for supporting this research.

exponential in the minimal width of such a refutation. In addition, they propose a simple dynamic programming procedure for automated theorem proving – one which simply searches for small width proofs. These results provide a theoretical motivation for understanding the problem of deciding whether or not a set of clauses has a proof of a given width.

However, there are also strong practical motivations from the areas of automated theorem proving and propositional reasoning for understanding this problem. For example, although SAT-solvers have been remarkably successful, there are some inputs on which they fail. Researchers working with SAT-solvers may therefore be tempted to design preprocessing algorithms which determine the width required to refute a given formula F. Since we know from [3] that short proofs are narrow, if the minimum width of F is sufficiently large, then any Resolution-based SAT-solver will fail in the case that F is unsatisfiable, so researchers would be able to tell ahead of time whether or not to bother trying to solve F, potentially saving a great deal of time and computer resources. Unfortunately, any hopes for such a preprocessing algorithm are dashed by our main result.

The width problem for resolution is as follows: given a set of clauses Σ, and integer k as input, determine whether or not there is a resolution refutation of Σ of width bounded above by k. This problem was conjectured to be $\mathcal{EXPTIME}$-complete by Moshe Vardi, a conjecture confirmed in the present paper. The proof proceeds by a reduction from the problem of determining the winner in the existential k-pebble game, recently proved $\mathcal{EXPTIME}$-complete by Kolaitis and Panttaja [7]. As an immediate consequence of our main result, we prove that the problem of determining the winner in the fixed template version of the existential k-pebble game is also $\mathcal{EXPTIME}$-complete.

2 Resolution Proofs and Their Width

A *clause* is a set of literals. We write the empty clause as $\emptyset$, use the notation $C \vee D$ for the clause $C \cup D$, and write $C \vee l$ for $C \cup \{l\}$, where l is a literal. We also employ the notation $x \leftrightarrow (y \vee z)$ as an abbreviation for the set of three clauses $\{\overline{x} \vee y \vee z, \overline{y} \vee x, \overline{z} \vee x\}$.

If $C \vee x$ and $D \vee \neg x$ are clauses, then the resolution rule allows us to derive the clause $C \vee D$, by *resolving on* the variable x. If Σ is a set of clauses, then a sequence of clauses $C_1, \ldots, C_k$ is a *resolution proof* of the clause C from Σ if every clause in the sequence either contains a clause in Σ, or is derived from earlier clauses in the sequence by resolution, and $C_k = C$; it is a *refutation* of Σ if $C = \emptyset$.

The *width* of a clause refers to the number of literals that it contains. The width $w(\Sigma)$ of a set of clauses is the maximum width of a clause in Σ, while the width of a resolution proof π is the maximum width of a clause in π. The *k-width resolution problem* is defined as follows. The input is a set of clauses Σ, and a number k; the problem is – does Σ have a resolution refutation of width no greater than k?

3 Characterization of Resolution Width

In this section, we give a proof of the result of Atserias and Dalmau characterizing the width of resolution refutations [1]. The characterization is in terms of a two-player game which we shall call the *k-width game*, and it is played as follows: The players are the *Prover* and the *Adversary*. [1] The game is played as follows.

The players are given a set of clauses Σ, on a set V of variables, and an integer parameter $k > 0$. The players together construct a succession of assignments to the variables V. Initially, the assignment is empty; at every round of the game, all assignments involve at most k variables. Each round of the game proceeds as follows. First, the Prover can delete zero or more of the variable assignments from a previous round. Second, the Prover queries a currently unassigned variable, and the Adversary assigns a value to it.

The Prover wins if the current assignment falsifies an initial clause in Σ. The Adversary wins if an earlier assignment is repeated during the play of the game.

Clearly every play of the game must eventually terminate with a win for the Prover or for the Adversary (Atserias and Dalmau define their game so that when the game continues infinitely, the Adversary wins). It follows that either the Prover or the Adversary must have a winning strategy.

Definition 3.1 *If Σ is a set of clauses on a set V of variables, then a non-empty family $\mathcal{F}$ of V-assignments is an* extendible k-family for Σ *if it satisfies the following conditions:*

1. *No assignment in $\mathcal{F}$ falsifies a clause in Σ;*
2. *If $\alpha \in \mathcal{F}$, and $\beta \subseteq \alpha$, then $\beta \in \mathcal{F}$;*
3. *If $\alpha \in \mathcal{F}$, $|\alpha| < k$, and $x \in V$, then there is a $\beta \in \mathcal{F}$, so that $\alpha \subseteq \beta$, and $\beta(x)$ is defined.*

The next theorem (Atserias and Dalmau [1]) shows that a resolution refutation of width k constitutes a winning strategy for the Prover, while an extendible $k+1$-family provides a winning strategy for the Adversary in the $k+1$-width resolution game.

Theorem 3.2 *Let Σ be a contradictory set of clauses, and $k \geq w(\Sigma)$. Then the following are equivalent:*

1. *There is no resolution refutation of Σ of width k;*
2. *There is an extendible $k+1$-family for Σ;*
3. *The Adversary wins the $k+1$-width game based on Σ.*

[1] Atserias and Dalmau, following the tradition of finite model theory, call their players the *Spoiler* and the *Duplicator*, but our terminology seems clearer in the present context.

Proof: First, let us suppose that there is no resolution refutation of Σ of width k. Define $\mathcal{C}$ to be the set of all clauses having a resolution proof from Σ of width at most k; by assumption, $\Sigma \subseteq \mathcal{C}$. Let $\mathcal{F}$ be the set of all assignments of size at most $k+1$ that do not falsify any clause in $\mathcal{C}$. We claim that $\mathcal{F}$ is an extendible $k+1$-family for Σ. First, $\mathcal{F}$ is non-empty, because it contains the empty assignment (since $\mathcal{C}$ does not contain the empty clause). Second, $\mathcal{F}$ satisfies the first two conditions of Definition 3.1, by construction. To prove the fourth condition, let $\alpha \in \mathcal{F}$, and $|\alpha| \leq k$, $x \in V$, but there is no extension β of α in $\mathcal{H}$ with $\beta(x)$ defined. It follows that there is a clause $D \in \mathcal{C}$ that is falsified if we extend α by setting x to 0. Then $D = E \vee x$ for some E, since otherwise α would falsify D. Similarly, there is a clause $F \vee \overline{x}$ in $\mathcal{C}$ that is falsified by the extension of α that sets x to 1. Then α must falsify $E \vee F$; but $E \vee F$ is in $\mathcal{C}$, contradicting our assumption.

Second, let us suppose that there is an extendible $k+1$-family $\mathcal{F}$ for Σ. Then the Adversary can play the $k+1$-width game on Σ by responding to the Prover's queries with the appropriate assignment from the family, starting with the empty assignment. Since no assignment in the family falsifies an initial clause, this strategy must eventually end in a win for the Adversary, no matter how the Prover plays.

Finally, let us suppose that there is a resolution refutation of Σ of width k. Then the refutation provides the Prover with a winning strategy in the $k+1$-width game based on Σ. Starting from the empty clause at the root, the Prover follows a path in the refutation to one of the leaves in the refutation. At each round, the current assignment (after appropriate deletions), is a minimal assignment falsifying a clause in the path. The variable queried is the variable resolved upon to derive the current clause, and the next clause in the path is one of the premises of the clause from the previous round. Since the refutation has width k, every assignment has size bounded by $k+1$, so this strategy must result in a win for the Prover. □

Corollary 3.3 *The k-width resolution problem is in $\mathcal{EXPTIME}$.*

Proof: On a given play of the game, it is possible to keep track of the number of current assignments that have appeared up to a given round, and so determine if a repetition has occurred. Since there are at most $N = \binom{n}{k}3^k$ possible assignments, where n is the number of variables in the clause set Σ employed in the game, when the count reaches $N+1$, a repetition must have occurred.

Consequently, the description of the game given above shows that there is an alternating Turing machine operating in polynomial space that determines whether the Prover or the Adversary wins a given instance of the game. Hence, the problem is in $\mathcal{EXPTIME}$. □

4 The Existential k-Pebble Game

The k-width resolution problem is a special case of the existential pebble game described in this section, as Atserias and Dalmau show in [1].

The *existential k-pebble game* [8], or $(\exists, k)$*-pebble game* for short, is played on two finite relational structures $\mathcal{A}$ and $\mathcal{B}$ of the same similarity type. A *partial homomorphism* φ between $\mathcal{A}$ and $\mathcal{B}$ is a mapping from a substructure of $\mathcal{A}$ to a substructure of $\mathcal{B}$ that preserves all of the relations in the structures; that is to say, for every relation $R^{\mathcal{A}}$ in $\mathcal{A}$, if $a_1, \ldots, a_m$ are in the domain of φ, and $R^{\mathcal{A}}(a_1, \ldots, a_m)$, then $R^{\mathcal{B}}(\varphi(a_1), \ldots, \varphi(a_m))$. A *homomorphism* between $\mathcal{A}$ and $\mathcal{B}$ is a partial homomorphism that is defined on all the elements of $\mathcal{A}$.

The $(\exists, k)$-pebble game, where $k \geq 2$ is a positive integer, is played by two players, the *Spoiler* and *Duplicator*, on the relational structures $\mathcal{A}$ and $\mathcal{B}$ (in the terminology of §3, the Prover plays the role of the Spoiler, the Adversary the role of the Duplicator). Each player has a set of k pebbles, numbered $1, \ldots, k$; we shall write $\{p_1, \ldots, p_k\}$ for the set of pebbles used by the Spoiler, and $\{q_1, \ldots, q_k\}$ for the set of pebbles used by the Duplicator. In each round of the game, the Spoiler can make one of two different moves: either removing a pebble p_i from a pebbled element of $\mathcal{A}$, or placing a free pebble p_j on an element of the domain of $\mathcal{A}$. To each move of the Spoiler, the Duplicator must respond, either by removing the corresponding numbered pebble q_i from an element of $\mathcal{B}$, or placing the pebble q_j on an element of $\mathcal{B}$.

The pebbles placed by the two players at any stage of the game define a relation R between the domains of $\mathcal{A}$ and $\mathcal{B}$; if a is an element of $\mathcal{A}$, and b of $\mathcal{B}$, then Rab holds if and only if there is a pebble p_i on a, and a corresponding pebble q_i on b. The Spoiler wins a play of the game at a given round if the relation R defined by the pebbling at that round is not a partial homomorphism between $\mathcal{A}$ and $\mathcal{B}$. The Duplicator wins if there is a repetition of an earlier position. As before, the game must terminate after a finite number of moves in a win for the Spoiler or the Duplicator, and so either the Spoiler or the Duplicator must have a winning strategy.

As in the case of the resolution game of §3, we can give a combinatorial characterization of a winning strategy for the Duplicator.

Definition 4.1 *Let $\mathcal{A}$ and $\mathcal{B}$ be two finite relational structures of the same similarity type. A non-empty family $\mathcal{H}$ of partial homomorphisms between $\mathcal{A}$ and $\mathcal{B}$ is an* extendible k-family *if it satisfies the following two conditions:*

1. $\mathcal{H}$ is closed under subfunctions: *If $h \in \mathcal{H}$, and $g \subseteq h$, then $g \in \mathcal{H}$.*
2. $\mathcal{H}$ has the k-extension property: *If $f \in \mathcal{H}$, $|f| < k$ and a is an element of $\mathcal{A}$, then there is an element b of $\mathcal{B}$ so that $f \cup \{\langle a, b\rangle\}$ is in $\mathcal{H}$.*

The characterization in the following theorem is the general result of which Theorem 3.2 is a special case.

Theorem 4.2 *Let $\mathcal{A}$ and $\mathcal{B}$ be two finite relational structures of the same similarity type, and k a positive integer. Then the following two statements are equivalent:*

1. *The Duplicator has a winning strategy for the $(\exists, k)$-pebble game on the structures $\mathcal{A}$ and $\mathcal{B}$.*

2. There is an extendible k-family of partial homomorphisms for $\mathcal{A}$ and $\mathcal{B}$.

Proof: See Kolaitis and Vardi [8, §4]. □

In the reduction of the next section, it is helpful to assume that the strategy for the Spoiler in the $(\exists, k)$-pebble game is of a restricted sort.

Lemma 4.3 *If there is a winning strategy for the Spoiler in the $(\exists, k)$-pebble game, then there is a strategy in which the Spoiler never places more than one pebble on any individual element of $\mathcal{A}$.*

Proof: If the Spoiler places a pebble p_j on an element of $\mathcal{A}$, where there is already a pebble p_i placed earlier in the game, then the Duplicator can respond by placing q_j on the same element of $\mathcal{B}$ as q_i. Any other response is an obvious blunder, since the relation defined from the resulting position is not a homomorphism. Consequently, such moves can give no advantage to the Spoiler, and so the Spoiler might as well play as if the Duplicator never makes such obvious blunders, and thus never place two pebbles on the same element of $\mathcal{A}$. □

5 Complexity of the k-Width Problem

In this section, we prove our main result by reducing the problem of determining the winner in the $(\exists, k)$-pebble game to the k-width problem for resolution. The former problem was proved $\mathcal{EXPTIME}$-complete by Kolaitis and Panttaja [7]. In fact, they prove the stronger result, that a special case of this problem is $\mathcal{EXPTIME}$-complete, a fact that is useful in our reduction.

A *coloured graph* is a relational structure $\mathcal{A}$ of the form $\langle A, E, C_1, \ldots, C_m \rangle$, where E is a symmetric, irreflexive relation on A, and $C_1, \ldots, C_m$ are subsets of A. Using this notion, we can state the main result of Kolaitis and Panttaja.

Theorem 5.1 *The problem of determining whether the Duplicator has a winning strategy in the $(\exists, k)$-pebble game on the structures $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are coloured graphs of the same similarity type, is $\mathcal{EXPTIME}$-complete under logspace reducibility.*

Proof: See Kolaitis and Panttaja [7]. □

We now give a reduction of the $(\exists, k)$-pebble game problem to the width problem for resolution by translating the first problem into a set of clauses in 3-CNF.

Definition 5.2 *Let $\mathcal{A} = \langle A, E, C_1, \ldots, C_m \rangle$ and $\mathcal{B} = \langle B, F, D_1, \ldots, D_m \rangle$ be coloured graphs, with $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$. For every i, $1 \leq i \leq p$, $\Sigma(\mathcal{A}, \mathcal{B})$ contains q variables P^i_j, for each j, $1 \leq j \leq q$, and in addition, $q - 1$ auxiliary variables Q^i_j, for $1 \leq j < q$. The clauses constituting $\Sigma(\mathcal{A}, \mathcal{B})$ are as follows:*

1. Q^i_1, for $1 \leq i \leq p$.

2. $Q^i_j \leftrightarrow (P^i_j \vee Q^i_{j+1})$, *for* $1 \leq j < q-1$, *and* $Q^i_{q-1} \leftrightarrow (P^i_{q-1} \vee P^i_q)$.

3. $\neg P^i_j$, *where* $a_i \in C_r$, $b_j \notin D_r$, *for some* r.

4. $\neg P^i_j \vee \neg P^s_t$, *where* $E(a_i, a_s)$, *but not* $F(b_j, b_t)$.

5. $\neg P^i_j \vee \neg P^i_k$, *where* $1 \leq j < k \leq q$.

The set of clauses $\Sigma(\mathcal{A}, \mathcal{B})$ is satisfiable if and only if there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$, and is logspace constructible from $\mathcal{A}$ and $\mathcal{B}$. If α is an assignment to the variables of $\Sigma(\mathcal{A}, \mathcal{B})$, we define $\mathrm{Dom}(\alpha)$ to be the set of all i for which $\alpha(P^i_j)$ is defined, for some j.

If $R \subseteq A \times B$, then we write $\alpha[R]$ for the assignment defined by: $\alpha[R](P^i_j) = 1$ if and only if $R(a_i, b_j)$. Furthermore, if f is a mapping from a subset of A to B, then we define the assignment $\beta[f]$ determined by f as follows. If $i \in \mathrm{Dom}(f)$, then $\beta[f](P^i_j) = 1$ if $f(a_i) = b_j$, and $\beta[f](P^i_j) = 0$ otherwise, while $\beta[f](Q^i_k) = 1$ if $f(a_i) = b_j$, for $k \leq j$, and $\beta[f](Q^i_k) = 0$ otherwise. It is easily checked that if f is a partial homomorphism from $\mathcal{A}$ to $\mathcal{B}$, $\beta[f]$ does not falsify any clause in $\Sigma(\mathcal{A}, \mathcal{B})$.

Lemma 5.3 *If* $\mathcal{A}$ *and* $\mathcal{B}$ *are coloured graphs, and* $k \geq 3$, *then the Spoiler has a winning strategy for the* $(\exists, k)$*-pebble game on* $\mathcal{A}$ *and* $\mathcal{B}$ *if and only if the Prover has a winning strategy for the* $k+2$*-width game on* $\Sigma(\mathcal{A}, \mathcal{B})$.

Proof: ($\Rightarrow$) First, assume that the Spoiler has a winning strategy for the $(\exists, k)$-pebble game on $\mathcal{A}$ and $\mathcal{B}$. By Lemma 4.3, we can assume that the strategy for the Prover never involves doubly pebbled elements; this implies that every relation R_t produced by the Spoiler's strategy is a map from a subset of A to B. Then the Prover has a winning strategy for the $k+2$-width game that follows the Spoiler's strategy. At each stage in the strategy, the current assignment α maintained by the Prover contains a set of at most k variables of the form P^i_j, all of them set to 1, and representing a partial map from A to B. In addition, α assigns values to at most two extra variables, each of these being either a variable P^i_q, or an auxiliary variable of the form Q^i_j.

The Prover's strategy consists of successively forcing the Adversary to assign the value 1 to variables of the form P^i_j, in such a way as to produce a series of assignments of the form $\alpha[R_0], \alpha[R_1], \ldots, \alpha[R_t], \ldots$, where $R_0, R_1, \ldots, R_t, \ldots$ are the relations $R_t \subseteq A \times B$ produced by following the Spoiler's strategy in the $(\exists, k)$-pebble game on $\mathcal{A}$ and $\mathcal{B}$.

Let us suppose that the Prover's current assignment is of the form $\alpha[R_t]$. If R_{t+1} is produced from R_t by removing pebbles, then the Prover simply deletes the appropriate variable assignment from $\alpha[R_t]$. So, let us suppose that the Spoiler places a free pebble on an element a_i of A, so that $|R_t| < k$. The Prover must now force the Adversary to set at least one variable P^i_j to 1.

The Prover begins by querying Q^i_1, which the Adversary is forced to set to 1; the Prover then queries P^i_q. If the Adversary assigns this last variable the value 1, then the Prover has succeeded. Otherwise, the Prover performs a binary

search in the sequence of variables $\sigma = Q^i_1, \ldots, Q^i_{q-1}, P^i_q$. At each stage in the search, the Prover retains two variables from σ in the current assignment, the first set to 1, the second set to 0. The next variable queried is chosen so as to cut the interval between the two variables in σ as nearly in half as possible. This search procedure must terminate with two consecutive variables in σ set to 1 and 0, respectively, say, Q^i_j and Q^i_{j+1}. Then the Adversary must assign 1 to the variable P^i_j on the next query, otherwise a clause of type 2 is falsified. Once the Prover has succeeded in forcing a response of this kind from the Adversary, the assignments to the extra variables are deleted, and the strategy continues with the new assignment of the form $\alpha[R_{t+1}]$, with $i \in \mathrm{Dom}(\alpha[R_{t+1}])$.

Since the strategy described above corresponds to a winning strategy for the Spoiler, any play using this strategy terminates in a win for the Prover, since it must end when one of the assignments $\alpha[R_t]$ falsifies a clause of type 3, 4 or 5 in Definition 5.2 (the Spoiler wins the $(\exists, k)$-pebble game when R_t is not a partial homomorphism from $\mathcal{A}$ to $\mathcal{B}$).

($\Leftarrow$) Second, assume that the Duplicator has a winning strategy for the $(\exists, k)$-pebble game on $\mathcal{A}$ and $\mathcal{B}$. We describe a winning strategy for the Adversary in the k-width game, based on the Duplicator's winning strategy. By Theorem 4.2, there is an extendible k-family $\mathcal{H}$ of homomorphisms for $\mathcal{A}$ and $\mathcal{B}$. At each round in a play of the k-width game, the Adversary has a partial homomorphism belonging to $\mathcal{H}$. Initially, this homomorphism is empty; the homomorphism is updated as the play proceeds. We now describe the Adversary's update procedure.

The Adversary plays so as to maintain the following properties invariant throughout the game:

1. f is a partial homomorphism in $\mathcal{H}$, and $|f| \leq k$.
2. $\{i : a_i \in \mathrm{Dom}(f)\} \subseteq \mathrm{Dom}(\alpha)$.

Let α be the assignment at the start of a given round of the k-width game, and $f \in \mathcal{H}$ the partial homomorphism that the Adversary has available at the end of the previous round. Initially, the Prover removes some variable assignments from α. If this results in the removal of some $i \in \mathrm{Dom}(\alpha)$, where $a_i \in \mathrm{Dom}(f)$, then f is restricted appropriately; that is to say, the new assignment f' is defined by $f' := f \restriction \{a_i : i \in \mathrm{Dom}(\alpha)\}$.

We now need to describe the variable querying part of a round. Let α be the current assignment maintained by the Prover, and f the partial homomorphism maintained by the Adversary. When the Prover queries an unset variable of the form P^i_j or Q^i_j, three cases arise:

1. If $a_i \in \mathrm{Dom}(f)$, then the Adversary answers in accordance with the assignment $\beta[f]$.
2. If $a_i \notin \mathrm{Dom}(f)$, but $|f| < k$, then the Adversary extends f to a new partial homomorphism $g \in \mathcal{H}$, with $a_i \in \mathrm{Dom}(g)$, and then replies according to $\beta[g]$.

3. If $a_i \notin \mathrm{Dom}(f)$, but $|f| = k$, then the Adversary sets any variable P_j^i to 0, and any variable Q_j^i to 1.

We need to prove that this strategy on the part of the Adversary succeeds. This amounts to showing that whenever the Adversary answers a query, that the resulting assignment never falsifies an initial clause in $\Sigma(\mathcal{A}, \mathcal{B})$. In the first two cases, this is clearly true, by the definition of an extendible k-family. Only the third case causes difficulties. Let α' be the extension of α after the Adversary's reply. In this case, there are at most two variables assigned values by α' that are not assigned values by $\beta[f]$. Since all such variables P_j^i are set to 0 by the Adversary, no clause of types 3,4 or 5 can be falsified by α', and since all such variables Q_j^i are set to 1, no clause of type 1 can be falsified. Finally, since at most two such variables are assigned values, no clause of type 2 can be falsified by α', completing the verification that the strategy used by the Adversary always succeeds in the $k+2$-width game using the set of clauses $\Sigma(\mathcal{A}, \mathcal{B})$. □

This brings us to our main result. We now show that the Resolution width problem is $\mathcal{EXPTIME}$-Complete. The language associated with this problem is formally defined as follows:

Definition 5.4 ($RES\text{-}WIDTH$) *$RES\text{-}WIDTH = \{(F, k) \mid F$ is a formula for which there exists a Resolution refutation with width space at most $k\}$*

Theorem 5.5 *$RES\text{-}WIDTH$ is $\mathcal{EXPTIME}$-complete under logspace reducibility.*

Proof: Theorem 5.1 of Kolaitis and Panttaja shows that the problem of determining the winner in an instance of the $(\exists, k)$-pebble game on coloured graphs $\mathcal{A}$ and $\mathcal{B}$ of the same similarity type is $\mathcal{EXPTIME}$-complete under logspace reducibility. Theorem 3.2 and Lemma 5.3 provide a logspace reduction of this problem to the $k+1$-width problem for $\Sigma(\mathcal{A}, \mathcal{B})$. □

6 Related Complexity Results

This chapter's main result in the previous section has an interesting corollary concerning another form of the $(\exists, k)$-pebble game:

Corollary 6.1 *The problem of determining the winner in an instance of the $(\exists, k)$-pebble game on relational structures $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{B}$ is a two-element structure, is $\mathcal{EXPTIME}$-complete under logspace reducibility.*

Albert Atserias has remarked that the preceding theorem also settles the complexity of the fixed template version of the existential k-pebble game. Feder and Vardi [4] (see also [2]) observe that the satisfiability problem for formulas in r-CNF can be encoded as a constraint satisfaction problem in the form where the target structure $\mathcal{B}$, or 'template' is fixed, while only the source structure, or 'instance' varies.

Corollary 6.2 *The problem of determining whether the Duplicator has a winning strategy in the $(\exists, k)$-pebble game on structures $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{B}$ is a fixed template on a two-element universe, is $\mathcal{EXPTIME}$-Complete under logspace reducibility.*

In addition, Theorem 5.5 has another Corollary which settles the k-width problem for Tree Resolution as well:

Corollary 6.3 *The k-width problem for Tree Resolution is $\mathcal{EXPTIME}$-complete under logspace reducibility.*

Proof: In order to prove this, it suffices to show that for any formula F, the Resolution width of refuting it is equal to its Tree Resolution width. In the forward direction, if we are given a Resolution refutation π of F with width at most k, then we can easily build a Tree Resolution refutation of F with width k by simply turning the DAG underlying the proof tree of π into a tree by duplicating subtrees as necessary in the obvious way. Clearly this does not introduce any new clauses, and therefore does not affect the proof width at all. The other direction is trivial, since any Tree refutation is a Resolution refutation. □

Acknowledgements

We would like to express our thanks to Steve Cook, who originally told us of the resolution width problem, and to Moshe Vardi for his stimulating conjecture, as well as his comments on an earlier draft. We would also like to thank Albert Atserias for his observation about the fixed template problem.

References

[1] A. Atserias and V. Dalmau. A Combinatorial Characterization of Resolution Width. *Proceedings of the* 18^{th} *IEEE Conference on Computational Complexity*, 2003.

[2] Albert Atserias. On Sufficient Conditions for Unsatisfiability of Random Formulas. *Journal of the Association for Computing Machinery*, 51:281–311, 2004. Preliminary version: 17^{th} IEEE Symposium on Logic in Computer Science (LICS), pages 275-284, 2002.

[3] E. Ben-Sasson and A. Wigderson. Short Proofs are Narrow -Resolution Made simple. *Journal of the Association for Computing Machinery*, 48:149–169, 2001. Preliminary version: Proceedings of the 31^{st} Annual ACM Symposium on Theory of Computing, 1999, pages 517-526.

[4] Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study Through Datalog and Group Theory. *Siam Journal on Computing*, 28:57–104, 1998.

[5] Z. Galil. On the Complexity of Regular Resolution and the Davis-Putnam Procedure. *Theoretical Computer Science*, 4:23 – 46, 1977.

[6] A. Haken. The Intractability of Resolution. *Theoretical Computer Science*, 39:297 – 308, 1985.

[7] Phokion G. Kolaitis and Jonathan Panttaja. On the Complexity of Existential Pebble Games. In *Proceedings of Computer Science Logic CSL '03*, pages 314–329. Springer, 2003. Lecture Notes in Computer Science 2803.

[8] Phokion G. Kolaitis and Moshe Y. Vardi. On the Expressive Power of Datalog: Tools and a Case Study. *Journal of Computer and System Sciences*, 51:110–134, 1995.

[9] G. S. Tseitin. On the Complexity of Derivation in Propositional Calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part 2*, pages 115 – 125. Consultants Bureau, New York, 1970.