Space Bounds for Resolution*

Juan Luis Esteban
Dept. L.S.I.
U. Politècnica de Catalunya
Jordi Girona Salgado 1-3
08023 Barcelona, Spain
esteban@lsi.upc.es

Jacobo Torán
Abt. Teoretische Informatik
Universität Ulm
Oberer Eselsberg
89069 Ulm, Germany
toran@informatik.uni-ulm.de

June 28, 2000

*Results included in this paper have appeared in the conferences STACS'99 [9] and CSL'99 [19]

Running head: Space bounds for Resolution

Contacting author:
Jacobo Torán
Abt. Theoretishe Informatik
Universität Ulm
Oberer Eselsberg
89069 Ulm, Germany
toran@informatik.uni-ulm.de

Abstract

We introduce a new way to measure the space needed in resolution refutations of CNF formulas in propositional logic. With the former definition [11] the space required for the resolution of any unsatisfiable formula in CNF is linear in the number of clauses. The new definition allows a much finer analysis of the space in the refutation, ranging from constant to linear space. Moreover, the new definition allows to relate the space needed in a resolution proof of a formula to other well studied complexity measures. It coincides with the complexity of a pebble game in the resolution graphs of a formula, and as we show, has relationships to the size of the refutation. We also give upper and lower bounds on the space needed for the resolution of unsatisfiable formulas. We show that Tseitin formulas associated to a certain kind of expander graphs of n nodes need resolution space $n - c$ for some constant c. Measured on the number of clauses, this result is the best possible. We also show that the formulas expressing the general Pigeonhole Principle with n holes and more than n pigeons, need space $n + 1$ independently of the number of pigeons. Since a matching space upper bound of $n + 1$ for these formulas exist, the obtained bound is exact. We also point to a possible connection between resolution space and resolution width, another measure for the complexity of resolution refutations.

1 Introduction and Definitions

In this paper we deal exclusively with propositional logic, and the only refutation system considered is resolution. Due to its simplicity and to its importance in automatic theorem proving and logic programming systems, resolution is one of the best studied refutation systems. Resolution contains only one inference rule: If $A \vee x$ and $B \vee \bar{x}$ are clauses, then the clause $A \vee B$ may be inferred by the resolution rule resolving the variable x. A resolution refutation of a conjunctive normal form (CNF) formula φ is a sequence of clauses $C_1 \ldots C_s$ where each C_i is either a clause of φ or is inferred from earlier clauses in the refutation by the resolution rule, and C_s is the empty clause, $\Box$. One way to measure the complexity of resolution applied to a specific formula, is to measure the minimum *size* of a refutation for it. This is defined as the number of clauses in the refutation. More than a decade ago, Haken [10] gave the first proof of an exponential lower bound on the number of clauses needed in any resolution refutation of a family of formulas expressing the pigeonhole principle. In following years, the original proof has been greatly simplified and extended to other classes of formulas [21, 6, 3, 18, 15].

Because of the importance of resolution, other measures for the complexity of resolution refutations have been introduced. Recently Ben-Sasson and Wigderson [4], building on previous work [3, 7], unified all the existing exponential lower bounds for resolution size using the concept of *width*. The width of a resolution refutation is the maximal number of literals in any clause of the refutation. The authors relate in [4] width and size showing that lower bounds for resolution width imply lower bounds for resolution size.

A less studied measure for the complexity of a resolution refutation is the amount of *space* it needs. This measure was defined in [11] in the following way:

Definition 1 *[11] Let $k \in I\!N$, we say that an unsatisfiable CNF formula φ has resolution refutation bounded by space k if there is a series of CNF formulas $\varphi_1, \ldots, \varphi_s$, such that $\varphi = \varphi_1$, $\Box \in \varphi_s$, in any φ_i there are at most k clauses, and for each $i < s$, φ_{i+1} is obtained from φ_i by deleting (if wished) some of its clauses and adding the resolvent of two clauses of φ_i.*

Intuitively this expresses the idea of keeping a set of *active* clauses in the refutation, and producing from this set a new one by copying clauses from the previous set and resolving one pair of clauses, until the empty clause is included in the set. Initially the set of active clauses consists of all the clauses of φ, and the space needed is the maximum number of clauses that are simultaneously active in the refutation.

In [11] it is proven that any unsatisfiable CNF formula φ with n variables and m clauses can be refuted in space $m + n$, and in [8] it is observed that the space upper bound $2m$ can also be obtained.

The above definition has the important drawback that the space needed in a refutation can never be less than the number of clauses in the formula being refuted. This is so because this formula is the first one in the sequence used to derive the empty clause. Making an analogy with a more familiar computation model, like the Turing machine, this is the same as saying that the space needed cannot be less than the size of the input being processed. To be able to study problems in which the *working space* is smaller than the size of the input, the space needed in the input tape is usually not taken into consideration. We do the same for the case of resolution and introduce the following alternative definition for the space needed in a refutation.

Definition 2 *Let $k \in \mathbb{N}$, we say that an unsatisfiable CNF formula φ has resolution refutation bounded by space k if there is a series of CNF formulas $\varphi_1, \ldots, \varphi_s$, such that $\varphi_1 \subseteq \varphi$, $\Box \in \varphi_s$, in any φ_i there are at most k clauses, and for each $i < s$, φ_{i+1} is obtained from φ_i by deleting (if wished) some of its clauses, adding the resolvent of two clauses of φ_i, and adding (if wished) some of the clauses of φ (initial clauses).*

The space needed for the resolution of an unsatisfiable formula is the minimum k for which the formula has a refutation bounded by space k.

In the new definition it is allowed to add initial clauses to the set of active clauses at any stage in the refutation. Therefore this clauses do not need to be stored and do not consume much space since in any moment at most two of them are needed simultaneously. The only clauses that consume space are the ones derived at intermediate stages. As we will see in Section 2, there are natural classes of formulas that can be refuted using only logarithmic space (in the number of initial clauses), or even constant space. Recently in [2] this definition of space for resolution has also been adopted.

There is another natural way to look at this definition using pebble games on graphs, a traditional model used for space measures in complexity theory and for register allocation problems (see [16]). Resolution refutations can be represented as directed acyclic graphs of in-degree two, in which the nodes are the clauses used in the refutation, and a vertex (clause) has outgoing edges to the resolvents obtained using this clause. In this graph the sources are the initial clauses, all the other nodes have in-degree two, and the unique sink is the empty clause. In case that in the refutation no derived clauses are reused, that is, when all the nodes (except maybe the sources) have out-degree one, the proof is called tree-like. There is a restriction of resolution called *regular* resolution in which is required that in every path from the empty clause to an initial clause in the refutation graph, every variable is solved at most once.

The space required for the resolution refutation of a CNF formula φ (as expressed in Definition 2) corresponds to the minimum number of pebbles needed in the following game played on the graph of a refutation of φ.

Definition 3 *Given a connected directed acyclic graph with one sink the aim of the pebble game is to put a pebble on the sink of the graph (the only node with no outgoing edges) following this set of rules:*

1) A pebble can be placed in any initial node, that is, a node with no predecessors.

2) Any pebble can be removed from any node at any time.

3) A node can be pebbled provided all its parent nodes are pebbled.

3') If all the parent nodes of node are pebbled, instead of placing a new pebble on it, one can shift a pebble from a parent node.

There are several variations of this simple pebble game in the literature. In fact, in [22] it is shown that the inclusion of rule 3' in the game can at most decrease by one the number of pebbles needed to pebble a graph, but in the worst case the saving is obtained at the price of squaring the number of moves needed in the game. We include rule 3' so that the number of pebbles coincides exactly with the space in Definition 2. This fact is stated in the following straightforward Lemma.

Lemma 4 *Let φ be an unsatisfiable CNF formula. The space needed in a resolution refutation of φ coincides with the number of pebbles needed for the pebble game played on the graph of a resolution refutation of φ.*

This second characterization of space in resolution proofs allow us to use techniques introduced for the estimation of the number of pebbles required for pebbling certain graphs, for computing the space needed in resolution refutations. However the estimation of the number of pebbles needed in the refutation of a formula is harder than the estimation of the number of pebbles needed for a graph, since in the first case one has to consider all the possible refutation graphs for the formula.

In Section 3 we give upper and lower bounds for the amount of space needed for resolution. When measuring the space relative to the number of variables in the initial formula we show that any unsatisfiable CNF formula with n variables has a resolution proof that uses space $n+1$, and we also obtain a matching lower bound, that is, we show that there are formulas on n variables whose refutation needs space $n+1$. We also obtain optimal space lower bounds for the two important families of Tseitin formulas and formulas expressing the Pigeonhole Principle.

Very similar results also hold for these families of formulas if the width instead of the space of a resolution refutation is used [4]. This is surprising since both measures seem unrelated, and suggest that there might be a relationship between the concepts of width and space. Space lower bounds for these families of formulas for resolution and Polynomial Calculus have been obtained independently in [2].

We show in Section 3.1 space lower bounds for the refutation of Tseitin formulas. This family of formulas was first defined by Tseitin [20], and express the principle that the sum of the degrees of the vertices in a graph must be even. Tseitin proved in [20] super-polynomial lower bounds on the size of regular resolution refutations for them. Later Urquhart [21] improved these bounds to exponential lower bounds for general resolution. We prove that the space needed for the resolution of a Tseitin formula with associated graph G is at least $ex(G) - \lfloor \frac{d}{2} \rfloor + 1$, where $ex(G)$ is the expansion of G and d its maximum degree. For Tseitin formulas corresponding to expander graphs with n nodes, this means that the space needed is at least $n - c$ for some constant c. These formulas have $O(n)$ variables and clauses, and because of the general space upper bound mentioned above, the space needed is $\Theta(n)$, and this linear lower bound on the number of initial clauses is optimal up to a constant factor[1].

The family of formulas for the general Pigeonhole Principle PHP_n^m express the fact that it is not possible to fit m pigeons in n pigeonholes (for $m > n$). As mentioned above, for the case $m = n + 1$, this was the first example of a family of formulas with an exponential resolution size lower bound [10]. We show that the negation of PHP formulas needs refutation space $n+1$, independently of the number of pigeons [2]. In this case we have an exact bound since Messner [13] has proven that $n + 1$ is also an upper bound for the space needed for the refutation of PHP formulas with n pigeonholes.

This lower bound result is also interesting due to the fact that the complexity of resolution refutations of the general Pigeon Hole Principle is not known. For example, only trivial lower bounds on the size are known when the number of pigeons m is greater than n^2. Buss and Pitassi [5] have shown that for the case of tree-like resolution, for any $m > n$, $\neg\mathrm{PHP}_n^m$ needs tree-like resolution refutation of size at least 2^n. This result can also be proven using a lower bound on the width of refutations for $\neg\mathrm{PHP}_n^m$ from [4]. Due to the fact that tree-like resolution refutations of size S require at most space $\lceil \log S \rceil + 1$, the above mentioned space lower bound for $\neg\mathrm{PHP}_n^m$ also provides the lower

[1]A linear space lower bound in the number of initial clauses for Tseitin formulas have been independently proven in [2]

[2]A $\Omega(n)$ lower bound for the resolution space of PHP_n^m have been obtained independently in [2]

bound 2^n on the size of tree-like resolution refutations for these formulas.

We obtain in Section 4 an upper bound on the size of a refutation of a formula in terms of the space needed for its resolution and the depth of the refutation. (The depth of a refutation is the size of the longest path from the empty clause to an initial clause in the refutation graph). We prove in Theorem 22 that if a formula φ has a resolution refutation of depth d that uses space s, then this refutation has size bounded by $\binom{d+s}{s}$. For types of resolution in which the depth of the proofs is bounded (like in the case of regular resolution), this provides an exponential upper bound for the resolution size in terms of the resolution space.

In the last section we study the space needed in tree-like refutations. We give a characterization of this measure in terms of lists of active clauses, and show then that for the case of tree-like resolution, the space needed in a refutation of a formula is at least as large as the refutation width minus the initial with of the formula. Again here we find a connection between the concepts of space and width.

2 Definitions and Examples

In this section we give two examples of families of unsatisfiable formulas that can be refuted within less space than its number of clauses. The first example are the formulas whose clauses are all possible combinations of literals in such a way that every variable appears once in every clause. We will see that the space needed to refute these formulas is bounded by the number of different variables in it. In fact we will prove a more general result about the space needed in a tree-like resolution.

Definition 5 *We say that a graph G_1 is embedded in a graph G_2 if a graph isomorphic to G_2 can be obtained from G_1 by adding nodes and edges or inserting nodes in the middle of edges of G_1.*

Observe that the number of pebbles needed for pebbling any graph is greater than or equal to the number of pebbles needed for pebbling any embedded subgraph in it. This is true since any pebbling strategy for the graph, also pebbles the embedded subgraph.

Let φ a CNF-formula, and α a (partial) truth assignment to the variables in φ. φ_α is a modification of φ according to α. For every variable x in α if its truth value is 1, all the clauses in φ containing the positive literal x are deleted and all occurrences of $\bar{x}$ are deleted. If the truth value of x is 0, then all clauses in φ containing $\bar{x}$ are deleted and all occurrences of the literal x are deleted.

The next lemma, an easy adaptation of [14, Theorem 1], states the well known fact that for a resolution refutation of a formula φ, for any partial truth assignment α to the variables, we can get a resolution refutation of φ_α, the formula after applying the partial assignment, embedded in the initial refutation.

Lemma 6 *Let Π be a resolution refutation of the CNF-formula φ, let α be a partial truth assignment and φ_α the formula after applying the partial assignment. There is a resolution refutation of φ_α whose resolution graph is embedded in Π.*

Proof. We construct a new refutation Π' transforming the clauses of Π. Every original clause is either eliminated or transformed into a new one. The new graph of clauses, after maybe contracting some adjacent nodes representing the same clause, is also a refutation graph, and by construction, the new refutation graph is embedded in the original one.

To build the new refutation we start transforming the initial clauses going downward following the original refutation. If an original clause contains a literal that has been assigned value 1 by α, then the whole clause is deleted. If it contains a literal with value 0, then the literal is deleted from the clause. Otherwise the clause remains unchanged.

If a clause in the original refutation is the resolvent of two previous ones, there are two cases depending on whether the resolved variable has been given a value by α or not. Suppose that clause C is the resolvent of $A \vee x$ and $B \vee \overline{x}$.

Case 1, variable x has been assigned by α. If $A \vee x$ (resp. $B \vee \bar{x}$) has been replaced by A' (resp. B') then C is replaced by A' (resp. B') if $\alpha(x) = 0$ (resp. $\alpha(x) = 1$).

Case 2, variable x has not been assigned by α. If $A \vee x$ (resp. $B \vee \bar{x}$) has been replaced by A' (resp. B') then C is replaced by the resolvent of A' and B' if both contain variable x, and otherwise C is replaced by any of A' or B' that does not contain variable x.

Consider the part of the new graph connected to the empty clause. Contracting nodes of in-degree one, we obtain a refutation graph that is embedded in the original one. ■

Theorem 7 *Let φ be an unsatisfiable CNF formula with a treelike resolution of size s, then φ has a resolution refutation of space $\lceil \log s \rceil + 1$.*

Proof. We will show that the resolution tree in the refutation of φ can be pebbled with $d+1$ pebbles, where d is the depth of the biggest complete binary tree embedded in the refutation graph. As the biggest possible complete binary tree embedded in a tree of size s has depth $\lceil \log s \rceil$, the theorem holds. It is a well known fact (see for example [16]) that $d+1$ pebbles suffice to pebble a complete binary tree of depth d (with the directed edges pointing to the root). In fact $d+1$ pebbles suffice to pebble any binary tree whose biggest embedded complete binary tree has depth d. In order to see this we use induction on the size of the tree. The base case is obvious. Let T be refutation tree, and T_1 and T_2 be the two subtrees from the root. Let us call $d_c(T)$ the depth of the biggest embedded subtree in T. So

$$d_c(T) = \begin{cases} \max(d_c(T_1), d_c(T_2)) & \text{if } d_c(T_1) \neq d_c(T_2) \\ d_c(T_1) + 1 & \text{if } d_c(T_1) = d_c(T_2) \end{cases}$$

By induction hypothesis one can pebble T_1 with $d_c(T_1) + 1$ pebbles and T_2 with $d_c(T_2) + 1$ pebbles. Let us suppose that $d_c(T_1) < d_c(T_2)$, then $d_c(T) = d_c(T_2)$ and one can pebble first T_2 with $d_c(T_2) + 1$ pebbles, leave a pebble in the root of T_2 and then pebble T_1 with $d_c(T_1) + 1$. For this second part of the pebbling one needs $d_c(T_1) + 2 \leq d_c(T_2) + 1$. The other case is similar. ■

We can apply the above lemma to compute the space needed in the refutation of the following formula.

Definition 8 *Let $n \in I\!N$,* COMPLETE-TREE$_n$ *is the CNF formula on the set of variables $\{x_1, \ldots, x_n\}$, whose clauses are all possible combinations of literals with the restriction that each variable appears once in each clause.*

$$\text{COMPLETE-TREE}_n = (x_1 x_2 \ldots x_n), (\bar{x}_1 x_2 \ldots x_n), \ldots, (\bar{x}_1 \bar{x}_2 \ldots \bar{x}_n).$$

Observe that this formula has 2^n clauses. It is not hard to see that COMPLETE-TREE$_n$ can be refuted using space $n+1$. This is so since a straightforward tree-like resolution of the formula that resolves the variables in different stages, has size $2^{n+1} - 1$. The previous lemma assures that this refutation can be pebbled with $n+1$ pebbles. In the next section we will see that this amount of space is also necessary.

As second example, consider the class of unsatisfiable formulas in CNF with at most two literals per clause.

Theorem 9 *Any unsatisfiable CNF formula with at most two literals in each clause can be resolved within constant space.*

Proof. The first part of the proof is similar to the one for showing that the set of 2-CNF unsatisfiable formulas can be recognized in nondeterministic logarithmic space.

In fact it is not hard to see that this result can also be derived from this Theorem. Given a 2-CNF formula φ one can construct a directed graph G_φ related to it. This graph will be useful to know whether the formula is unsatisfiable or not, and in the former case, will provide us with a strategy to find a refutation that can be pebbled with constant space.

The set V of vertices of G_φ is the set of literals in φ. For any clause $(x_1 \vee x_2)$ (that can be viewed as the implication $\bar{x}_1 \rightarrow x_2$ or also $\bar{x}_2 \rightarrow x_1$) we include in E a directed edge from $\bar{x}_1$ to x_2 and another one from $\bar{x}_2$ to x_1. If the clause has only one literal x_1 we consider it as $(x_1 \vee x_1)$ and include in E and edge from $\bar{x}_1$ to x_1. No other edge is included in E.

The formula is unsatisfiable if and only if there is a cycle in the graph that contains a literal, say x_1, and its negation. We can use this cycle to get a resolution refutation. Starting from node x_1, let us call the clauses related to the edges in the cycle $C_1, C_2, \ldots, C_k$ (all these are initial clauses of the refutation), and suppose that $C_1, \ldots C_l$ are the clauses corresponding to the edges from x_1 to $\bar{x}_1$ in the cycle, and $C_{l+1} \ldots C_k$ correspond to the edges from $\bar{x}_1$ to x_1. One can resolve C_1 with C_2 getting a new clause which will be resolved with C_3 and so on. When resolving with C_l one gets the clause x_1. For this only 2 pebbles are needed. Analogously, starting from literal $\bar{x}_1$ one can resolve C_{l+1} with C_{l+2} and so on, until resolving with C_k and thus getting the clause $\bar{x}_1$. Resolving finally both clauses x_1 and $\bar{x}_1$ the empty clause is obtained. This shows that at most 3 pebbles are needed to pebble such a refutation. ■

3 Upper and Lower Bounds

For the results in this section the following concept will be very useful.

Definition 10 *We say that a CNF unsatisfiable formula is minimally unsatisfiable if removing any clause the formula becomes satisfiable.*

The following result attributed to M. Tarsi can be found in [1].

Lemma 11 *Any minimally unsatisfiable CNF formula must have more clauses than variables.*

We start by giving bounds with respect to the number of variables.

Theorem 12 *Every unsatisfiable formula with n variables can be resolved using resolution in space at most $n+1$.*

Proof. As mentioned in the proof of Theorem 7, for pebbling a tree of depth d, $d+1$ pebbles suffice. If we consider regular tree-like resolution, which is complete, we have refutation trees whose depth is at most the number of variables in the formula being refuted. ■

There is a matching lower bound, since there are formulas of n variables whose refutation graphs can only be pebbled with $n+1$ pebbles. This is a consequence of the following result:

Theorem 13 *Let φ an unsatisfiable CNF formula and k the smallest number of literals of a clause of φ. Any resolution refutation of φ needs at least space $k+1$.*

Proof. For any pebbling strategy, there is a first step, let us call it s, in which the set of pebbled clauses becomes unsatisfiable. This step must exist because the first pebbling step consists of pebbling an initial clause, which is always satisfiable, and the last step pebbles the empty clause.

In step s, an initial clause has to be pebbled since according to the pebbling rules the only other possibility would be to pebble a clause with both parents pebbled, and this step would not transform the set of pebbled clauses into an unsatisfiable set. Therefore the set of pebbled clauses at step s contains at least k variables (the ones of the initial clause).

Let us suppose than the set of pebbled clauses at step s is minimally unsatisfiable, then, by Lemma 11, it has at least $k+1$ clauses because it has at least k variables. On the other hand, if this set is not minimally unsatisfiable, we can throw aside clauses until the remaining set becomes minimally unsatisfiable. Notice that we cannot delete the initial clause last added to the set, otherwise the set of clauses would be a subset of the clauses at stage $s-1$ and becomes therefore satisfiable. So, $k+1$ clauses are still needed because the initial clause is contained in the set and has at least k variables. ■

Since all the clauses in COMPLETE-TREE$_n$ have n variables, we obtain:

Corollary 14 *For all $n \in \mathbb{N}$ any resolution refutation of* COMPLETE-TREE$_n$ *requires at least space $n+1$.*

Theorem 13 can be strengthened to allow to prove lower bounds for the space needed in the refutation of a more general class of formulas.

Theorem 15 *Let φ be a unsatisfiable CNF formula, and let k be the maximum over all partial assignments α of the minimum number of literals of a clause in φ_α. The space needed in a resolution refutation of φ is at least k.*

Proof. Let α be any partial assignment to the variables in φ, and Π a refutation of φ that needs the smallest amount of space. From Lemma 6 we know that there exists a refutation Π' for φ_α embedded in the structure of Π. Theorem 13 guarantees that to pebble φ_α one needs at least a number of pebbles equal to the length of the shortest clause in φ_α. But as Π' is embedded in Π, one cannot pebble Π with fewer pebbles than Π'. To finish the proof we just need to consider an assignment α which produces a shortest clause of maximal length. ■

3.1 Lower bounds on Tseitin formulas

In this section we study the space used in resolution refutations of some formulas related to graphs. These formulas were defined originally by Tseitin [20], and have also been used in order to prove lower bounds on the size of resolution refutations in [21] and [18].

Let $G = (V, E)$ be a connected undirected graph with n vertices, and let $m : V \to \{0, 1\}$ be a marking of the vertices of G satisfying the property

$$\sum_{x \in V} m(x) = 1 (\text{mod } 2).$$

For such a graph we can define an unsatisfiable formula in conjunctive normal form $\varphi(G, m)$ in the following way: The formula has E as set of variables, and is a conjunction of the translation in CNF of the formulas φ_x for $x \in V$, where

$$\varphi_x = \begin{cases} e_1(x) \oplus \ldots \oplus e_d(x) & \text{if} \quad m(x) = 1 \\ \overline{e_1(x) \oplus \ldots \oplus e_d(x)} & \text{if} \quad m(x) = 0 \end{cases}$$

Here $e_1(x) \ldots e_d(x)$ are the edges (variables) incident with vertex x. If d is the maximum degree of a node in G, $\varphi(G, m)$ contains at most $n2^{d-1}$ many clauses, each one with at most d many literals. The number of variables of the formulas is bounded by $\frac{dn}{2}$.

$\varphi(G, m)$ captures the combinatorial principle that for all graphs the sum of the degrees of the vertices is even. When the marking m is odd, $\varphi(G, m)$ is unsatisfiable. Suppose on the contrary that there were a satisfying assignment $\alpha : E \to \{0, 1\}$. For every vertex x, the number of edges of x that have been assigned value 1 by α has the same parity as $m(x)$, and therefore

$$\sum_{x \in V} \sum_{(x,y) \in E} \alpha((x, y)) \equiv \sum_{x \in V} m(x) \equiv 1 (\text{ mod } 2)$$

but in the left hand sum in the equality, every edge is counted twice and therefore this sum must be even, which is a contradiction.

The following fact was also used in [21] and [18] and plays a fundamental role in the proof of the lower bound. For completeness we include a proof of it.

Fact 16 *For an odd marking m, for every $x \in V$ there exists an assignment α with $\alpha(\varphi_x) = 0$, and $\alpha(\varphi_y) = 1$ for all $y \neq x$. If the marking is even, then $\varphi(G, m)$ is satisfiable.*

Proof. Let m be an odd marking and x be a node in V. The desired assignment α can be constructed in the following way: We start with an assignment β with $\beta(\varphi_x) = 0$. For an odd number of nodes v (including x) the value of φ_v under β in now 0. We pick two such nodes u and v different from x. Since the graph is connected there must be a path from u to v. We toggle in the assignment β the value of all the variables (edges) along this path. Now the values of φ_u and φ_v are also changed and these formulas have therefore value 1. On the other hand, the formulas related to the nodes w lying between u and v keep the same truth value as before because for these formulas the truth value of two of the edges has been changed. This procedure is repeated for different pairs of nodes u, v until the desired assignment α is found. For an even marking m the proof is completely analogous. ■

Consider a a partial truth assignment α of some of the variables. We refer to the following process as applying α to (G, m): Setting a variable (x, y) in α to 0 corresponds to deleting the edge (x, y) in the graph, and setting it to 1 corresponds to deleting the edge from the graph and toggling the value of $m(x)$ and $m(y)$ in G. Observe that the formula $\varphi(G', m')$ for the graph and marking (G', m') resulting after applying α to (G, m) is still unsatisfiable.

In order to prove the lower bound we will consider the last stage in any pebbling strategy in which two properties are satisfied. On the one hand, the set of pebbled clauses must be simultaneously satisfiable. The other property needed is based on non-splitting assignments, a concept that we define next.

Definition 17 *We say that a partial truth assignment α of some of the variables in $\varphi(G, m)$ is non-splitting for (G, m), if applying it to (G, m) produces a pair (G', m') so that G' has a connected component of size $> \frac{2}{3}n$ with an odd number of 1's in its marking, and an even number of 1's in the markings of all other connected components.*

Definition 18 *Let $G = (V, E)$ be an undirected graph with with $|V| = n$. The expansion of G, $ex(G)$ is defined as:*

$$ex(G) = \min k: \ \exists S \subseteq V, |S| \in \left[\frac{n}{3}, \frac{2n}{3}\right], \ |\{(x, y) \in E: \ x \in S, y \notin S\}| = k.$$

Intuitively the expansion of a graph is the minimum size of a cut produced when the vertices are partitioned into two subsets that do not differ too much in size. As shown in the next theorem, the expansion of a graph is a lower bound on the space required in the resolution of its associated Tseitin formula.

Theorem 19 *Let $G = (V, E)$ be an undirected and connected graph with $|V| = n$ and maximum degree d, and let m be an odd marking of G. Any resolution refutation of $\varphi(G, m)$ requires space at least $ex(G) - \lfloor \frac{d}{2} \rfloor + 1$.*

Proof. Let Π be a resolution refutation of the formula, and consider the last stage s in a pebbling strategy of the graph of Π in which there is a partial assignment α fulfilling the following two properties:

i) α simultaneously satisfies all the pebbled clauses at stage s,

ii) α is *non-splitting* for (G, m).

This stage in the pebbling must exist: Before the initial step, no clause has a pebble. Since G is connected, the empty truth assignment is trivially a non-splitting partial assignment satisfying the set of pebbled clauses. At the end, the set of pebbled clauses contains the empty clause which cannot be satisfied by any assignment. Stage s must exist in between.

The clause pebbled in stage $s+1$ must be an initial one. The only other clause that could be pebbled at stage $s+1$ would be a clause C_3 whose parents C_1 and C_2 already have a pebble, but any partial assignment satisfying C_1 and C_2 also satisfies C_3, and the non-splitting partial assignment from stage s would also work for stage $s+1$. For some vertex x in G, this last initial pebbled clause corresponds to the formula φ_x.

Let α be a partial assignment satisfying properties *i)* and *ii)* at stage s. There is an extension of α that satisfies φ'_x, the formula for x after applying α. To see this, observe that after applying α to (G, m), the graph has a connected component of size at least $\frac{2n}{3}$ with an odd marking, and the rest of the components have even markings. By Fact 16, for every vertex x, the formula φ'_x can therefore be satisfied by an extension of α. Moreover, the initial clause C pebbled at stage $s+1$ corresponds to a vertex x in the big connected component with odd marking since otherwise there would be also non-splitting partial assignments satisfying all the pebbled clauses at stage $s+1$.

Let α be a non-splitting partial truth assignment of minimal size satisfying the clauses at stage s, and (G', m') the graph and marking resulting after applying α. It suffices to extend α giving some value to one or more of the variables in the last pebbled clause to obtain an assignment α' satisfying all the clauses pebbled at stage

$s+1$. However, α' is a splitting assignment and applying it to (G,m) does not produce a connected component larger than $\frac{2}{3}n$ with odd marking. We will show that there is always a way to extend α to α' by assigning some new variables in the last pebbled clause C, in such a way that α' satisfies all the pebbled clauses and produces a subgraph disconnected from the rest and with a number of nodes in the interval $[\frac{n}{3}, \frac{2n}{3}]$

Let C be the initial clause pebbled at stage $s+1$, corresponding to a node x and let d' be the degree of x in G' ($d' \leq d$). $\varphi'(x)$ is the formula $e_1(x) \oplus \ldots \oplus e_{d'}(x) = m'(x)$. We have shown that this formula is satisfiable. d' is at least 1, since otherwise α would also satisfy φ'_x.

x is connected in G' to d' components $A_1, \ldots, A_{d'}$, and there is no edge between any two of such components A_i, A_j. Otherwise, satisfying the clause C by satisfying the literal corresponding to the edge connecting x and A_i, would provide a non-splitting extension of α.

We consider different cases depending on the size of the A components.

Case 1: Some component A_i has size within the interval. Deleting the edge connecting x and A_i, this component is isolated from the rest of the graph.

Case 2: The size of all the A_i components lie outside the interval. This implies that they all have size smaller than $\frac{n}{3}$, since otherwise, by Fact 16, there would be an extension of α that satisfies C, and disconnects all the components form node x producing an odd marking in the component of size greater that $\frac{2n}{3}$, and an even marking in all the other ones. This would provide a non-splitting assignment satisfying all the pebbled clauses at stage $s+1$. The size of all the components A_i is therefore smaller than $\frac{n}{3}$ and the sum of all their sizes is greater than $\frac{2n}{3}$. There is a set of at most $\lfloor \frac{d'}{2} \rfloor$ components such that the sum of their sizes lie within the interval. This set of components can be isolated from the rest of the graph just by deleting the edges connecting them to x.

In both cases, by deleting at most $\lfloor \frac{d}{2} \rfloor$ edges from G' we have isolated a set of nodes S of size within $[\frac{n}{3}, \frac{2n}{3}]$ from the rest of the graph. There are at least $ex(G)$ edges $\{y,z\}$ in G with $y \in S$ and $z \notin S$. All these edges, except at most $\lfloor \frac{d}{2} \rfloor$ of them have been removed by the partial assignment α. Since α was chosen to be an assignment of minimal size satisfying all the pebbled clauses at stage s, there are at least $ex(G) - \lfloor \frac{d}{2} \rfloor$ pebbled clauses at this stage and $ex(G) - \lfloor \frac{d}{2} \rfloor + 1$ pebbled clauses at stage $s+1$. ■

There exist expander graphs G with n nodes constant degree d and with $ex(G) > n$ [12]. In [17] it is shown that the degree for such expander graphs can be reduced to $d = 8$. For an odd marking of such a graph the formula $\varphi(G,m)$ has at most $\frac{dn}{2}$ variables and $n2^{d-1}$ clauses. By the above result, the space needed in a resolution refutation of $\varphi(G,m)$ is at least $n-3$ as stated in the next corollary:

Corollary 20 *For the constant $d = 8$ there is a family of unsatisfiable formulas $\varphi_1, \varphi_2, \ldots$ (corresponding to expander graphs) such that for every n φ_n has at most $256n$ clauses and $4n$ variables, and any resolution refutation of φ_n requires at least space $n - 3$.*

The number of variables of a formula is an upper bound for its resolution space (Theorem 12). For the family of formulas mentioned in the corollary, the space needed is therefore $\Theta(n)$. Observe that this bound is linear, measured in terms of the number of clauses of the formula.

An interesting fact is that Theorem 19 (even with the lower bound $ex(G)$ instead of $ex(G) - \lfloor \frac{d}{2} \rfloor + 1$) also holds if the width of the refutation instead of the space is considered [4].

3.2 The Pigeonhole Principle

Let $m > n$. The tautology PHP^m_n expresses the Pigeonhole Principle that there is no one-one mapping from a domain of size m (the set of pigeons) into a range of size n (the set of holes). We study the space needed in a resolution refutation of the contradiction $\neg\mathrm{PHP}^m_n$. This contradiction can be written as a CNF formula in the following way: The variables of the formula are $x_{i,j}$, $1 \leq i \leq m, 1 \leq j \leq n$. $x_{i,j}$ has the intuitive meaning that pigeon i is mapped to hole j. There are mn variables. The clauses of the formula are:

(1) $x_{i,1} \vee x_{i,2} \vee \ldots \vee x_{i,n}$ for $1 \leq i \leq m$, and

(2) $\overline{x}_{i,k} \vee \overline{x}_{j,k}$ for $1 \leq i, j \leq m,\ 1 \leq k \leq n,\ i \neq j$.

Clauses of type (1) express the fact that every pigeon is mapped to some hole, while the clauses of type (2) indicate that at most one pigeon can be mapped to any hole.

The number of clauses in $\neg\mathrm{PHP}^m_n$ is $m + \binom{m}{2}n < m^2 n$.

Theorem 21 *For any $m > n$, the space needed in a resolution refutation of $\neg PHP^m_n$ is at least $n + 1$.*

Proof. Let Π be a resolution refutation of $\neg\mathrm{PHP}^m_n$ and consider the last stage s in a pebbling strategy of the graph of Π in which there is a partial assignment α fulfilling the following two properties:

i) α simultaneously satisfies all the pebbled clauses at stage s, and

ii) α does not assign value false to any of the initial clauses.

At stage $s = 0$ in the pebbling process, such a partial assignment α exists since there are no pebbled clauses. Also, at the end of the pebbling, the empty clause has a pebble on it and therefore there is no α fulfilling property *i*). Because of this, the stage s defined above must exist.

The pebble from stage $s+1$ is placed in an initial clause. Otherwise the two parents of the pebbled clause at stage $s+1$ contain a pebble in stage s and any partial assignment satisfying the pebbled clauses at stage s also satisfies the clauses at stage $s+1$.

Let α be a partial assignment simultaneously satisfying all the pebbled clauses at stage s. α can be extended to a partial assignment α' that satisfies the last pebbled clause C. We have seen that C must be an initial clause. If no extension of α can satisfy clause C it is because α assigns value false to all the literals in C, but this is a contradiction since C is an initial clause, and by condition *ii*) α cannot give value false to any initial clause.

Let α be a partial assignment of minimal size satisfying all the pebbled clauses at stage s and not giving value false to any initial clause, and let α' be any extension of α satisfying the clause C pebbled at stage $s+1$. By hypothesis, α' falsifies some initial clause.

If C is of type (1) for some pigeon i, C can be satisfied by giving value true to some variable $x_{i,k}$ that has not been assigned by α. This makes some initial clause $C_{i,k}$ false, and therefore $C_{i,k}$ must be of type (2), $C_{i,k} = \overline{x_{i,k}} \vee \overline{x_{j,k}}$ for some j. This implies that for any hole k, α assigns variable $x_{i,k}$ value false, or variable $x_{j,k}$ value true (for some $j \neq i$), and therefore α assigns at least as many variables as holes. Since α was a partial assignment of minimal size satisfying all the pebbled clauses at stage s, in this stage at least n clauses were pebbled, and in $s+1$ at least $n+1$.

If C is of type (2), $C = \overline{x_{i,k}} \vee \overline{x_{j,k}}$, assigning value true to any literal in C that has not been assigned by α, falsifies some initial clause of type (1). If α has not assigned value to any of the variables in C, this means that the number of variables assigned by α is at least $2n-2$. Otherwise α has assigned at least n variables. For $n \geq 2$, this implies that the number of variables assigned by α is at least n, which means that the number of pebbled clauses at stage $s-1$ is at least n, and at stage s, $n+1$. ■

Jochen Messner [13] has proved that $n+1$ pebbles suffice in a resolution refutation of the Pigeonhole Principle with n holes and $m > n$ pigeons. This means the the above space lower bound is exact.

Although only trivial lower bounds for the size of a resolution refutation of the general Pigeonhole Principle $\neg\mathrm{PHP}^m_n$ are known for the case $m > n^2$, the situation is better when restricted to tree-like resolution. In [5] it is shown that for any $m > n$,

$\neg\mathrm{PHP}^m_n$ requires tree-like resolution refutations of size 2^n. Using Theorem 7 we can derive this bound as a corollary of the above space lower bound. The same bound for tree-like refutations of $\neg\mathrm{PHP}^m_n$ has also obtained in [4] using a lower bound on the width of the refutations of $\neg\mathrm{PHP}^m_n$.

4 Relationships between Space and Size

The main result of this section provides an upper bound on the size of resolution refutations of a formula in terms of the space and the depth needed in a refutation[3]. Recall that the depth of a resolution refutation is the size of the longest path from the empty clause to an initial clause in the graph of the refutation.

Theorem 22 *A resolution refutation for an unsatisfiable CNF formula φ on n variables using space s and depth d, has size at most $\binom{d+s}{s}$.*

Proof. Let Π be the resolution refutation proof that can be pebbled with s pebbles. The depth of a clause C in Π is the length of the longest path from C to the empty clause.

We associate a set A of at most s clauses in Π with an array $\mathrm{depth}(A) = a_1 \ldots a_s$ of s numbers between 1 and $d+1$ in the following way: Sort the clauses in A by depth in Π and for $1 \le j \le s$ let a_j be the depth of the clause of j-th smallest depth. If there are less than j clauses in A then let $a_j = d+1$. In this way the array $\mathrm{depth}(A)$ has always s positions. We can compare these arrays as base $d+1$ numbers in the usual way.

Π can be pebbled with s pebbles. W.l.o.g. we can suppose that in the pebbling strategy pebbles are removed from clauses in the first moment they are not needed anymore, that is, pebbles can only be removed from a clause only immediately after one of its successors has been pebbled.

In the pebbling strategy pebbles are placed and removed. We consider the stages right before the pebbles are placed. Let φ_i be the set of clauses containing pebbles at the stage right before the i-th time a pebble is set or shifted. φ_1 is the empty set. Observe that, by the special form of pebbling strategy we are considering, φ_{i+1} is obtained from φ_i by pebbling one clause, and eventually removing one or the two predecessors of this clause.

We claim that if φ_{i+1} and φ_i are two consecutive pebbling stages as described, then $\mathrm{depth}(\varphi_i) > \mathrm{depth}(\varphi_{i+1})$. If in stage φ_{i+1} no clauses are deleted, then the result is

[3]In [9] a better upper bound was announced. Unfortunately, the proof of the mentioned result is incorrect, and it is not known whether the result holds.

clear, since either one of the non-used pebbles at stage i (with depth $d+1$) is placed at depth $\leq d$, or some pebble is shifted to a position with smaller depth. In the other case one or two pebbles are deleted in stage $i+1$, but this can only happen if at stage $i+1$ a clause C resolvent of the clauses with the removed pebbles is pebbled. φ_{i+1} differs from φ_i since it contains C and does not contain one or the two predecessors of C. Since the depth of C is smaller than the depth of its predecessors the inequality holds.

In each stage i in the pebbling strategy at most a new clause is considered and it holds $\text{depth}(\varphi_i) > \text{depth}(\varphi_{i+1})$. Because of this the number of clauses in the refutation is bounded by the set of possible values of the function $\text{depth}(A)$ for sets A of size at most s. $\text{depth}(A)$ is encoded by an ordered sequence of s numbers ranging from 1 to $d+1$. Since there are $\binom{d+s}{s}$ possible values for these sequences, the size of the refutation is bounded by $\binom{d+s}{s}$. ■

We get several consequences from this result:

Corollary 23 *Any family of unsatisfiable CNF formulas with resolution refutations of polynomial depth and constant space, have resolution refutations of polynomial size.*

In some types of resolution, the depth of the proof is automatically bounded. An example is regular resolution. For this type of resolution it is required that in every path from the empty clause to an initial clause in the refutation graph, every variable is solved at most once. Clearly in this case the number of variables is a bound on the depth of the proof.

Corollary 24 *If an unsatisfiable CNF-formula on n variables has a regular resolution refutation of space s, then the size of this refutation is bounded by $\binom{n+s}{s}$.*

An interesting question is whether the depth of the refutation can be taken out of the bound given by Theorem 22. A way to do this would be by showing that a refutation of a formula can be transformed into another one that uses the same amount of space, but has bounded depth. It is not clear that this result holds, but as we see in the next section, it does hold for the case of tree-like resolution.

5 Space in tree-like resolution

We consider in this section the question of measuring the space when the resolution refutations are restricted to be tree-like. Recall that in this case all the nodes in the underlying graph have fan out one, and that the same clause may appear more than once in this graph. Since in definition 2 does not refer to the structure of the underlying graph, we measure initially the tree-like space needed for the refutation of an unsatisfiable formula as the minimum number of pebbles needed to play the game on a refutation tree of the formula. Later on we will show that it is also possible to give a characterization of tree-like space in terms of list of clauses kept in memory, in a similar way as in definition 2. We start showing that a tree-like resolution can be made regular without increasing the space. Tseitin [20] showed that the same result holds also if the size of the refutation tree (instead of the space) is considered.

Theorem 25 *If φ is a CNF unsatisfiable formula with a tree-like resolution refutation that can be pebbled with s pebbles, then φ has a tree-like regular resolution refutation with the same amount of pebbles.*

Proof. Let φ be any formula and Π any tree-like refutation of φ and for any clause C let T_C be the subtree in the refutation tree that derives C from initial clauses. Suppose that the last resolution step in the refutation (the one having $\square$ as resolvent) resolves the variable x, and that this variable is resolved more than once in R. Applying Lemma 6 to T_x (resp. $T_{\bar{x}}$) with the partial truth assignment $\alpha(x) = 0$ (resp. $\alpha(x) = 1$) and then adding again the literal x (resp. $\bar{x}$) to the clauses that had it deleted, one derives x (resp. $\bar{x}$) or directly the empty clause. Putting both refutation trees together, the resulting tree-like refutation is embedded in Π and resolves variable x at most once. One can continue in this way with the parent clauses of x and $\overline{x}$ modifying the refutation until the initial clauses are reached. The way in which the new refutation is constructed assures that on every path from the empty clause to an initial one, every variable is resolved at most once, and moreover the new refutation in embedded in the former one, and therefore it does not need anymore space. ■

We can give now a definition of space in tree-like resolution considering list of clauses kept in memory, with the particularity that when a clause is used to derive other clauses, it is removed from the memory.

Definition 26 *Let $k \in I\!N$, we say that an unsatisfiable CNF formula φ has a tree-like resolution refutation bounded by space k if there is a series of CNF formulas (without having repeated clauses) $\varphi_1, \ldots, \varphi_s$, such that $\varphi_1 \subseteq \varphi$, $\square \in \varphi_n$, in any φ_i there are at most k clauses, and for each $i < s$, φ_{i+1} is obtained from φ_i by*

- *deleting (if wished) some of its clauses,*
- *adding the resolvent of two clauses of φ_i and deleting the parent clauses.*

- *adding (if wished) some of the clauses of φ (initial clauses).*

We show the equivalence of this definition and the one using pebbles. Clearly if a formula can be refuted in space k according to definition 26, then there is a refutation tree than can be pebbled with k pebbles.

For the other direction, the successive lists φ_i will be formed by the pebbled clauses in the tree. A problem can happen in case there are repetitions in the set of pebbled clauses, because in the list there can be only one copy of each clause. When deleting one instance of this clause we are deleting the only occurrence of the clause in the list. We show that one can always have a tree-like refutation using the same space and in which two occurrences of the same clause are never pebbled simultaneously.

Lemma 27 *Let s be the minimum number of pebbles needed in any tree-like refutation of φ. There is a regular tree-like resolution refutation of φ that can be pebbled with s pebbles in such a way that two nodes corresponding to the same clause are not pebbled simultaneously.*

Proof. By Theorem 25 we can suppose that there is a tree-like regular refutation of φ using s pebbles. Since every clause in the tree has at most one successor clause, when the successor clause is pebbled, in any sensible strategy, the parent clause can be deleted immediately. In Theorem 7 it is proved that the space needed to pebble a tree is the depth of its biggest embedded subtree. An optimal strategy is then: starting from the root, pebble first the subtree with the biggest embedded complete subtree and then the other subtree. Apply this rule recursively to both subtrees. If we follow this strategy when a clause, A is pebbled then we pebble the subtree that derives its mating clause A'. Since we are dealing with a regular refutation, A cannot be in the tree deriving A'. Otherwise, there would a path going from the copy of A deriving A' to the resolvent of A and A' and then to the empty clause, in which a variable has to be resolved twice, contradicting the fact that we are dealing with regular resolution. ■

Using Theorem 25 and the fact that in the proof of Theorem 22, applies to any kind of resolution, we get:

Corollary 28 *If an unsatisfiable formula φ with n variables has a tree-like resolution refutation of space s, then it has a tree-like resolution refutation of size $\binom{n+s}{s}$.*

The relationship between the two complexity measures of space and width is not clear. Recall that width of a refutation denotes the maximum number of literals of a clause appearing in the refutation. Formally:

Definition 29 *[4] The width of a clause C, $w(C)$, is defined as the number of literals in C. The width of a set of clauses in the maximal width of a clause in the set. The width of deriving a clause C from the formula φ, denoted $w(\varphi \vdash C)$ is defined by $min_{\Pi}\{w(\Pi)\}$ where the minimum is taken over all resolution derivations Π of C from φ.*

In the case of tree-like resolution we can show a connection between the concepts of size and width. For any unsatisfiable formula φ, the difference between the width in a refutation of φ minus the initial width of the formula, is bounded by the space in any tree-like refutation of the formula. The proof of this fact relies on the following lemma from Ben-Sasson and Wigderson:

Lemma 30 *[4] Let φ be a CNF unsatisfiable formula, and for a literal a, let φ_0 and φ_1 be the formulas resulting from assigning a the truth values 0 and 1 respectively. If for some value k, $w(\varphi_0 \vdash \Box) \leq k-1$ and $w(\varphi_1 \vdash \Box) \leq k$ then $w(\varphi \vdash \Box) \leq max\{k, w(\varphi)\}$*

Corollary 31 *Tree-space$(\varphi) - 1 \geq w(\varphi \vdash \Box) - w(\varphi)$.*

Proof. Let φ be an unsatisfiable CNF formula, and s the minimum number of pebbles needed in any tree-like refutation of φ, Π. We prove by induction on the depth of Π, d, that $w(\varphi \vdash \Box) \leq w(\varphi) + s - 1$. For $d = 0$, we have that $\Box$ is an initial clause, and the results holds trivially. For $d > 0$, let Π be a tree-like refutation of φ of depth d and let x be the last variable being resolved. Let T_0 and T_1 be the subtrees in the refutation deriving the literals x and $\overline{x}$ from initial clauses, and let s_0 and s_1 be the number of pebbles needed to pebble these subtrees reaching the literals x and $\overline{x}$.

Since we are dealing with a tree-like refutation, by (the proof of) Theorem 7, either s_0 or s_1 must be smaller than s. W.l.o.g. let us consider $s_0 < s$. Also, T_0 and T_1 have depth smaller than d.

Applying the partial assignment $x = 0$ to all the clauses in T_0 (respectively the partial truth assignment $x = 1$ to the clauses in T_1), we obtain two refutation trees deriving the empty clause from two sets of clauses φ_0, φ_1. By induction, $w(\varphi_0 \vdash \Box) \leq w(\varphi_0) + s_0 - 1 \leq w(\varphi) + s - 2$, and $w(\varphi_1 \vdash \Box) \leq w(\varphi_1) + s_1 - 1 \leq w(\varphi) + s - 1$. Applying Lemma 30 we obtain $s - 1 \geq w(\varphi \vdash \Box) - w(\varphi)$ ■

This result shows that width lower bounds can be used to obtain space lower bounds for the restricted case of tree-like resolution. Consider for example, for the case of a Tseitin formulas related to an undirected graph G with odd marking. Ben-Sasson and Wigderson have proved a width lower bound of the expansion of G [4]. By Corollary 31, this can be translated into a space lower bound for tree-like resolution of this formulas of at least the expansion of G minus the maximal degree of the graph. This is a little worse than the space lower bound for general resolution from theorem 3.1.

6 Conclusions and Open Problems

We have introduced a new definition to measure the space needed in the resolution of an unsatisfiable formula. This definition is more natural than the former one since it is closer to space measures in other complexity models and can be characterized in terms of a well studied pebble game. We have obtained upper and matching lower bounds for the space needed, as well as relationships between the space and the size of a refutation. These results bring new insight in the structure of resolution and hopefully will be useful in the analysis of refutations. Besides the interest the bounds have on their own for a better understanding of the studied classes of formulas, some of these results point to a possible connection between the seemingly unrelated measures of resolution width and space. Similar lower bounds to the ones shown here, hold also for the case of width, and besides, it is known that for the case of tree-like resolution both width and space lower bounds imply exponentially larger size lower-bounds. It has been shown in [2] that this is not true for the case of general resolution. However, the question of whether space lower bounds imply size lower bounds for other restrictions of resolution is still open.

There are several other interesting problems that remain open, like for example whether Theorem 22 can be modified so that the depth is not a parameter in the upper bound for the size, or whether it is true that every unsatisfiable formula that can be resolved in logarithmic space, has a resolution refutation of polynomial size (an improvement of Corollary 23).

Acknowledgment: The authors would like to thank an anonymous referee for simplifying Theorem 22 and Jochen Messner for helpful discussions on earlier versions of the paper.

References

[1] Aharoni, R., Linial, N.: Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. Journal of Combinatorial Theory, **43** (1986) 196–204.

[2] Alekhnovich, M., Ben-Sasson, E., Razborov A. and Wigderson, A.: Space complexity in Propositional Calculus. In *Proc. 32nd ACM Symp. on Theory of Computing* (2000) 358–367.

[3] Beame, P. and Pitassi, T.: Simplified and Improved Resolution Lower Bounds. In *Proc. 37th IEEE Symp. on Foundations of Computer Science*, (1996) 274–282.

[4] Ben-Sasson, E. and Widgerson, A.: Short proofs are narrow, resolution made simple. In *Proc. 31st ACM Symp. on Theory of Computing* (1999) 517–527.

[5] Buss, S. and Pitassi, T.: Resolution and the Weak Pigeonhole Principle. In *Proc. Computer Science Logic 97*, Springer Verlag LNCS 1414 (1997) 149–156.

[6] Chvátal, V. and Szemerédi, E.: Many hard examples for resolution. *Journal of the ACM*, **35** (1988) 759–768.

[7] Clegg, M., Edmonds, J. and Impagliazzo, R.: Using the Groebner Basis algorithm to find proofs of unsatisfiability. In *Proc. 28th ACM Symp. on Theory of Computing* (1996) 174–183.

[8] Esteban, J.L.: Complejidad de la resolución en espacio y tiempo. Masters thesis. Facultad de Informática de Barcelona. 1995.

[9] Esteban, J.L. and Torán, J.: Space bounds for resolution. In *Proc. 16th STACS*, Springer Verlag LNCS **1563** (1999) 551–561.

[10] Haken, A: The intractability of resolution *Theoretical Computer Science* **35** (1985) 297–308.

[11] Kleine Büning, H. and Lettman, T.: *Aussangenlogik: Deduktion und Algorithmen*, B.G. Teubner Stuttgart (1994).

[12] Margulis, A.: Explicit construction of concentrators. *Problems Information Transmission* **9** (1973) 71–80.

[13] Messner, J.: Personal communication 1999.

[14] Pudlák, P.: Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. The Journal of Symbolic Logic. **62, 3** (1997) 981–998.

[15] Razborov, A. Wigderson, A. and Yao, A. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In *Proc. 29th ACM Symp. on Theory of Computing* (1997) 739–748.

[16] Savage, J.: Models of Computation. Addison-Wesley. 1998.

[17] Schöning, U.: Better expanders and superconcentrators by Kolmogorov complexity. In *Proceedings 4th International Colloquium on Structural Information and Communication Complexity, Sirocco 97* Carleton Scientific (1997) 138–151.

[18] Schöning, U.: Resolution proofs, exponential bounds and Kolmogorov complexity. In *Proc. 22nd MFCS Conference*, Springer Verlag LNCS **1295** (1997) 110–116.

[19] Torán, J.: Lower bounds for the space used in Resolution. In *Proc. Computer science Logic Conference*, Springer Verlag LNCS 1683 (1999) 362–373.

[20] Tseitin, G.S. On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic*, Part 2. Consultants Bureau (1968) 115–125.

[21] Urquhart, A.: Hard examples for resolution. *Journal of the ACM* **34** (1987) 209–219.

[22] van Emde Boas, P., van Leeuwen, J.: Move rules and trade-offs in the pebble game. In Proc. 4th GI Conference, Springer Verlag LNCS. **67** (1979) 101–112.