Electronic Colloquium on Computational Complexity, Report No. 170 (2018)

Cops-Robber games and the resolution of Tseitin formulas

Nicola Galesi[1], Navid Talebanfard[2], and Jacobo Torán[3]

[1]Sapienza Università di Roma
[2]Czech Academy of Sciences
[3]Universität Ulm

October 4, 2018

Abstract

We characterize several complexity measures for the resolution of Tseitin formulas in terms of a two person cop-robber game. Our game is a slight variation of the one Seymour and Thomas used in order to characterize the tree-width parameter. For any undirected graph, by counting the number of cops needed in our game in order to catch a robber in it, we are able to exactly characterize the width, variable space and depth measures for the resolution of the Tseitin formula corresponding to that graph. We also give an exact game characterization of resolution variable space for any formula.

We show that our game can be played in a monotone way. This implies that the associated resolution measures on Tseitin formulas correspond exactly to those under the restriction of Davis-Putnam resolution, implying that this kind of resolution is optimal on Tseitin formulas for all the considered measures.

Using our characterizations we improve the existing complexity bounds for Tseitin formulas showing that resolution width, depth and variable space coincide up to a logarithmic factor, and that variable space is bounded by the clause space times a logarithmic factor.

1 Introduction

Tseitin propositional formulas for a graph $G = (V, E)$ encode the combinatorial statement that the sum of the degrees of the vertices of G is even. Such formulas provide a great tool for transforming in a uniform way a graph into a propositional formula that inherits some of the properties of the graph. Tseitin formulas have been extensively used to provide hard examples for resolution or as benchmarks for testing SAT-solvers. To name just a few examples, they were used for proving exponential lower bounds on the minimal size required in tree-like and regular resolution [17], in general resolution [18] and for proving lower bounds on resolution proof measures as the width [7] and the space [9], or more recently for proving time-space trade-offs in resolution [5, 6]. Due to the importance of these formulas, it is of great interest to find ways to understand how different parameters on the underlying graphs are translated as some complexity measures of the corresponding Tseitin formula. This was the key of the mentioned resolution results. For example the expansion of the graph translated into resolution lower bounds for the corresponding formula in all mentioned lower bounds, while the carving-width or the cut-width of the graph were used to provide upper bounds for the resolution width and size in [2, 5].

In this paper we obtain an exact characterization of the complexity measures of resolution width, variable space and depth for any Tseitin formula in terms of a cops-robber game played on its underlying graph.

ISSN 1433-8092

There exists a vast literature on such graph searching games (see eg. [10]). Probably the best known game of this kind is the one used by Seymour and Thomas [15] in order to characterize exactly the graph tree-width parameter. In the original game, a team of cops has to catch a robber that moves arbitrarily fast in a graph. Cops and robber are placed on vertices, and have perfect information of the positions of the other player. The robber can move any time from one vertex to any other reachable one but cannot go through vertices occupied by a cop. Cops are placed or removed from vertices and do not move. The robber is caught when a cop is placed on the vertex where she is standing. The value of the game for a graph G is the minimum number of cops needed to catch the robber on G. In [15] Seymour and Thomas also showed that this game is monotone in the sense that there is always an optimal strategy for the cops in which they never occupy the same vertex again after a cop has been removed from it. In a previous version of the game [13] the robber is invisible and the cops have to search the whole graph to be sure to catch her. The minimum number of cops needed to catch the robber in this game on G, characterizes exactly the path-width of G [8]. The invisible cop game is also monotone [13].

Our game is just a slight variation from the original game from [15]. The only differences are that the cops are placed on the graph edges instead of on vertices, and that the robber is caught when she is completely surrounded by cops. We show that the minimum number of cops needed to catch a robber on a graph G in this game, exactly characterizes the resolution width of the corresponding Tseitin formula. We also show that the number of times some cop is placed on some edge of G exactly coincides with the resolution depth of the Tseitin formula on G. Also, if we consider the version of the game with an invisible robber instead, we exactly obtain the resolution variable space of the Tseitin formula on G.

We also show that the ideas behind the characterization of variable space in terms of a game with an invisible robber, can in fact be extended to define a new combinatorial game to exactly characterize the resolution variable space of any formula (not necessarily a Tseitin formula). Our game is a non-interactive version of the Atserias and Dalmau game for characterizing resolution width [4].

An interesting consequence of the cops-robber game characterizations is that the property of the games being monotone can be used to show that for the corresponding complexity measures, the resolution proof can have the Davis-Putnam property (DP for short) without changing the bounds. This means that the relative order in which the variables are resolved is the same in every resolution path. As mentioned, the vertex-cops games are known to be monotone. This did not need to be true for our game. In fact, the robber-marshals game [11], another version of the game in which the cops are placed on the (hyper)edges, is know to be non-monotone [1]. We are able to show that the edge-cops game (for both cases of visible and invisible robber) is also monotone. This is done by reducing our edge game to the Seymour and Thomas vertex game. Using this fact and new game characterizations of the complexity measures for Davis-Putnam resolution we show that in the context of Tseitin formulas, the width, variable space and depth measures in Davis-Putnam resolution proofs are as good as in general resolution. A long standing open question from Urquhart [18] asks whether regular resolution can simulate general resolution on Tseitin formulas (in size). Since DP resolution is a restricted version of regular resolution, our results show that this is true for the measures of width and variable space[1].

Finally we use the game characterization to improve the known relationships between different complexity measures on Tseitin formulas. In particular we show that for any graph G with n vertices, the resolution depth of the corresponding formula is at most its resolution width times $\log n$. From this follows that all the three measures width, depth and variable space are within a logarithmic factor in Tseitin formulas. Our results provide a uniform class of propositional formulas where clause space is polynomially bounded in the variable space. No such result was known before as recently pointed to by Razborov in [14].

[1] The resolution depth is well known to coincide with the regular resolution depth for any formula.

The paper is organized as follows. In Section 2, we have all the necessary preliminaries on resolution and its complexity measures. We also provide in this section new versions of the Spoiler-Duplicator game from [4] for the cases of regular and DP-resolution width. In Section 3 we present the new game characterization of variable space in resolution. In Section 4 we introduce our variants of the Cops-Robber games on graphs and we show the characterizations of width, variable space and depth of the Tseitin formula on G in terms of Cops-Robber games played on G. In Section 5 we focus on the monotone version of the games and we prove that all our characterizations can be made monotone. In the last Section 6 we use our previous results to prove the new relationships between width, depth, variable space and clause space for Tseitin formulas. The last section contains some conclusions and open questions.

Finally in the Appendix, as a reading map of the paper, we include a series of Tables describing and referencing all the measures we consider and all the results we obtain in the paper.

2 Preliminaries

Let $[n] = \{1, 2, ..., n\}$. A *literal* is either a Boolean variable x or its negation $\bar{x}$. A *clause* is a disjunction (possibly empty) of literals. The empty clause will be denoted by $\Box$. The set of variables occurring in a clause C, will be denoted by $\mathsf{Vars}(C)$. The *width* of a clause C is defined as $\mathsf{W}(C) := |\mathsf{Vars}(C)|$.

A CNF F_n over n variables $x_1, \ldots, x_n$ is a conjunction of clauses defined over $x_1, \ldots, x_n$. We often consider a CNF as a set of clauses and to simplify the notation in this Section we omit the index n expressing the dependencies of F_n from the n variables. The width of a CNF F is $\mathsf{W}(F) := \max_{C \in F} \mathsf{W}(C)$. A CNF is a k-CNF if all clauses in it have width at most k.

An *assignment* for a set of variables X, specifies a truth-value ($\{0, 1\}$ value) for all variables in X. Variables, literals, clauses and CNFs are simplified under partial assignments (i.e. assignment to a subset of their defining variables) in the standard way.

The *resolution* proof system is a refutational propositional system for CNF formulas handling with clauses, and consisting of the only *resolution rule*:

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D}$$

A *proof* π of a clause C from a CNF F (denoted by $F \vdash_\pi C$) is a sequence of clauses $\pi := C_1, \ldots, C_m$, $m \geq 1$ such that $C_m = C$ and each C_i in π is either a clause of F or is obtained by the resolution rule applied to two previous clauses (called *premises*) in the sequence. When C is the empty clause $\Box$, π is said to be a *refutation* of F. Resolution is a sound a complete system for unsatisfiable formulas in CNF.

Let $\pi := C_1, \ldots, C_m$ be a resolution proof from a CNF F. The *width* of π is defined as $\mathsf{W}(\pi) := \max_{i \in [m]} \mathsf{W}(C_i)$. The width needed to refute an unsatisfiable CNF F in resolution is $\mathsf{W}(F \vdash) := \min_{F \vdash_\pi \Box} \mathsf{W}(\pi)$. The *size* of π is defined as $\mathsf{S}(\pi) := m$. The size needed to refute an unsatisfiable CNF F in resolution is $\mathsf{S}(F \vdash) := \min_{F \vdash_\pi \Box} \mathsf{S}(\pi)$.

Resolution proofs $F \vdash_\pi C$, can be represented also in two other notations: as directed acyclic graphs (DAG) or as sequences of set of clauses $\mathbb{M}$, called *(memory) configurations*. As a DAG, π is represented as follows: source nodes are labeled by clauses of F, the (unique) target node is labeled by C and each non-source node, labeled by a clause D, has two incoming edges from the (unique) nodes labeled by the premises of D in π. Using this notation the size of a proof π, is the number of nodes in the DAG representing π. The DAG notation allow to define other proof measures for resolution proofs. The *depth* of a proof π, $\mathsf{D}(\pi)$ is the length of the longest path in the DAG representing π. The depth for refuting an unsatisfiable CNF F is $\mathsf{D}(F \vdash) := \min_{F \vdash_\pi \Box} \mathsf{D}(\pi)$.

The representation of resolution proofs as configurations was introduced in [9, 3] in order to define *space* complexity measures for resolution proofs. A proof π, $F \vdash_\pi C$, is a sequence $\mathbb{M}_1, \ldots, \mathbb{M}_s$ such that: $\mathbb{M}_1 = \emptyset$, $C \in \mathbb{M}_s$ and for each $t \in [s-1]$, $\mathbb{M}_{t+1}$ is obtained from $\mathbb{M}_t$, by one of the following rules:

[*Axiom Download*]: $\mathbb{M}_{t+1} = \mathbb{M}_t \cup \{D\}$, for D a clause in F;

[*Erasure*]: $\mathbb{M}_{t+1} \subset \mathbb{M}_t$;

[*Inference*]: $\mathbb{M}_{t+1} = \mathbb{M}_t \cup \{D\}$, if $A, B \in \mathbb{M}_t$ and $\frac{A \quad B}{D}$ is a valid resolution rule.

π is a refutation if C is $\square$.

The *clause space* of a configuration $\mathbb{M}$ is $\mathsf{Cs}(\mathbb{M}) := |\mathbb{M}|$. The clause space of a refutation $\pi := \mathbb{M}_1, \ldots, \mathbb{M}_s$ is $\mathsf{Cs}(\pi) := \max_{i \in [s]} \mathsf{Cs}(\mathbb{M}_i)$. Finally the clause space to refute an unsatisfiable F is $\mathsf{Cs}(F \vdash) := \min_{F \vdash_\pi \square} \mathsf{Cs}(\pi)$. Analogously, we define the *variable space* and the *total space* of a configuration $\mathbb{M}$ as $\mathsf{Vs}(\mathbb{M}) := |\bigcup_{C \in \mathbb{M}} \mathsf{Vars}(C)|$ and $\mathsf{Ts}(\mathbb{M}) := \sum_{C \in \mathbb{M}} \mathsf{W}(C)$. Variable space and total space needed to refute an unsatisfiable F, are respectively $\mathsf{Vs}(F \vdash) := \min_{F \vdash_\pi \square} \mathsf{Vs}(\pi)$ and $\mathsf{Ts}(F \vdash) := \min_{F \vdash_\pi \square} \mathsf{Ts}(\pi)$.

2.1 Regular and Davis-Putnam resolution

We will consider two restrictions of general resolution. We say that a resolution refutation π of F is *regular* if in its graph representation, each variable is resolved at most once in each path from an initial clause of F to the empty clause. Analogously a resolution derivation π of a clause C is called regular if each variable is resolved at most once in each path in the derivation and no variable in C is resolved in π.

A regular resolution refutation is called *Davis-Putnam* if there is a total ordering of the variables so that in each path in the graph representation of the refutation, the variables are resolved relative to this order.

The complexity measures mentioned above, can be considered in the context of regular or Davis-Putnam resolution. In some cases it is easier to use the graph model for this. For example the regular (Davis-Putnam) resolution size of F is the minimum number of clauses in a regular (Davis-Putnam) resolution refutation graph of F, and similarly for the width measure. For measures defined in terms of configuration sequences, like variable space, some additional definition is needed.

Definition 1. *A* regular configuration refutation *for F is a configuration sequence $\mathbb{M}_1, \ldots \mathbb{M}_s$ refuting F with the following additional conditions: Each clause C in each configuration carries as additional information the set S_C of variables that have been resolved in the proof in order to derive C. For a clause C in F, $S_C = \emptyset$. If C is the result of resolving variable x from clauses D and E, then $S_C = S_D \cup S_E \cup \{x\}$. The refutation is regular if no clause C in the refutation contains a variable $x \in S_C$.*

Definition 2. *The variable space of a regular configuration refutation π is the maximum number of variables being present in any configuration of π. The regular variable space of an unsatisfiable formula F,* $\mathsf{regVs}(F \vdash)$*, is the minimum variable space in any regular configuration refutation of F.*

Definition 3. *We can define a Davis-Putnam configuration refutation of a formula F in the same way as with the regular configuration refutation, but with the additional requirement that there is an ordering σ of the variables and these are resolved in the relative order of σ. This means that for any clause C in the refutation, if C is the result of resolving a variable x, then x comes later than the rest of the variables in S_C in the order defined by σ. Analogously the variable space of a Davis-Putnam configuration refutation π is the maximum number of variables being present in any configuration of π and the DP variable space of an unsatisfiable formula F,* $\mathsf{dpVs}(F \vdash)$*, is the minimum variable space in any DP configuration refutation of F.*

2.2 Tseitin formulas

Let $G = (V, E)$ be a connected undirected graph with n vertices, and let $\varphi : V \to \{0, 1\}$ be an *odd* marking of the vertices of G, i.e. satisfying the property

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

For such a graph we can define an unsatisfiable formula in conjunctive normal form $\mathsf{T}(G, \varphi)$ in the following way: The formula has E as set of variables, and it is a conjunction of the CNF translation of the formulas F_x for $x \in V$, where F_x expresses that $e_1(x) \oplus \cdots \oplus e_d(x) = \varphi(x)$ and $e_1(x) \ldots e_d(x)$ are the edges (variables) incident with vertex x.

$\mathsf{T}(G, \varphi)$ encodes the combinatorial principle that for all graphs the sum of the degrees of the vertices is even. $\mathsf{T}(G, \varphi)$ is unsatisfiable if and only if the marking φ is odd. For an undirected graph $G = (V, E)$, let $\Delta(G)$ denote its maximal degree. It is easy to see that $\mathsf{W}(\mathsf{T}(G, \varphi)) = \Delta(G)$.

The following fact was proved several times (see for instance [9, 18]).

Fact 1. *For an odd marking φ, for every $x \in V$ there exists an assignment α_φ such that $\alpha_\varphi(F_x) = 0$, and $\alpha_\varphi(F_y) = 1$ for all $y \neq x$. Moreover if φ is an even marking, then $\mathsf{T}(G, \varphi)$ is satisfiable.*

Consider a partial truth assignment α of some of the variables of $\mathsf{T}(G, \varphi)$. We refer to the following process as applying α to (G, φ): Setting a variable $e = (x, y)$ in α to 0 corresponds to deleting the edge e in the graph G, and setting it to 1 corresponds to deleting the edge from the graph and toggling the values of $\varphi(x)$ and $\varphi(y)$ in G. Observe that $\mathsf{T}(G, \varphi)$ is satisfiable if and only if the formula $\mathsf{T}(G', \varphi')$ resulting after applying α to (G, m) is still unsatisfiable.

2.3 Spoiler Duplicator Games

In order to characterize the width of refuting Tseitin formulas in Resolution through the Cops-Robber game, we use another game introduced by Atserias and Dalmau in [4]. We consider the simplified explanation of the game from [16]. The *Spoiler-Duplicator* (SD-Game) is a two player game played on an $CNF\ F$. The players are given the set of clauses in F, with variables V. The players together construct a set of partial assignments to the variables in V, according to the following rules: At the begging the first assignment is λ, the empty assignment. At each round, on a given current partial assignment α, Spoiler can *select* an unassigned variable x or *forget* (unassigning) a variable from α. In the first case the Duplicator assigns a value to x, in the second case she does not do anything. The value of a game is the maximum number of variables that are assigned in α at some point during the game. Spoiler wins if the current assignment falsifies a clause in F, which will be always possible on an unsatisfiable CNF. The value of a given game on F, is the maximal number of variables simultaneously assigned at any point during the game. We define $\mathsf{sd}(F)$ as the minimum possible value of a terminating game on F. Using this game in [4] the authors provided the following characterization of with:

Theorem 1. *([4]) Let F be an unsatisfiable CNF. Then* $\mathsf{W}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{sd}(F) - 1\}$[2]

Following Urquhart [19] we define a Spoiler-Duplicator game tailored to capture the width in regular resolution. Since in this kind of resolution variables can be resolved in the proof only once in any path to

[2] In the original paper [4] it is stated that $\mathsf{W}(F \vdash) = \mathsf{sd}(F) - 1$, by inspecting the proof it can be seen that the formulation $\mathsf{W}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{sd}(F) - 1\}$ is the correct one.

the empty clause, we have to identify variables that were forgotten. Therefore variables can be in 3 *states*: (1) *assigned*; (2) *unassigned* and (3) *queried*. The *Regular-SD game* is played on CNF F exactly as the SD-game: at each round the Spoiler selects an *unassigned* variable and Duplicator gives it a value changing its state to *assigned*. If a variable is forgotten then its state become *queried*. A queried variable cannot be queried again. The Spoiler wins if the current assignment falsifies a clause in F. The value of a given regular SD-game on F, is the maximal number of variables queried and assigned at round in the game. We define $\mathsf{rsd}(F)$ as the minimum possible value of a terminating regular-SD-game on F.

Urquhart observed that the *Regular-SD* game characterizes exactly the width in regular resolution. However this proof is incorrect.[3] We consider here a different proof of this result.

Theorem 2. *Let F be an unsatisfiable CNF. Then* $\mathsf{regW}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{rsd}(F) - 1\}$

Proof. We show first that for an unsatisfiable CNF, $\mathsf{rsd}(F) \leq \mathsf{regW}(F \vdash) + 1$. Consider a regular resolution refutation of width k of F. A strategy for Spoiler is to start at the empty clause in the refutation and move towards a clause in F. In each step, he queries the variable being resolved, and depending on Spoiler's answer moves to the parent clause that is negated by the partial assignment constructed so far. He then deletes from the partial assignment all the variables that are not in the actual clause. Proceeding this way, a clause in F is negated and Spoiler needs at most $k+1$ variables in memory. Also since the refutation is regular, Spoiler does not need to ask any variable more than once.

For the other direction, let $0 < k = \mathsf{regW}(F \vdash)$ and suppose $k > \mathsf{W}(F)$, we give a strategy for Duplicator under which $\mathsf{rsd}(F) \geq k+1$.

A Spoiler-Duplicator game proceeds in rounds. In each round r Spoiler chooses a variable and Duplicator assigns a value to it or Spoiler forgets a variable. Let $V(r)$ be the set of variables that have been chosen by Prover until the end of round r (some of this variables might have been forgotten). Initially $V(0) = \emptyset$. Let α_r be the partial assignment of the variables at the end of round r, and let $|\alpha_r|$ be the size of domain of this partial assignment. We show that if for every round r, $|\alpha_r| \leq k$ then there is a strategy for Duplicator that never falsifies a clause in F. This implies $\mathsf{rsd}(F) \geq k+1$. We need one more definition. For a round r let $\mathcal{C}_r$ be the set of all clauses C satisfying:

$i)$ $\mathsf{W}(C) \leq k-1$,

$ii)$ C can be derived from F using regular resolution of clause width at most $k-1$ and

$iii)$ in the regular resolution proof of C no variable in $V(r)$ is resolved.

Observe that for all r, $F \subseteq \mathcal{C}_r$ since $\mathsf{W}(F) \leq k-1$. The strategy for Duplicator is to assign a value for the variable chosen by Spoiler at round r, that does not falsify a clause in $\mathcal{C}_r$. We claim that this is always possible. Observe that if the claim is true then this is clearly a winning strategy for Duplicator since the clauses in F are contained in $\mathcal{C}_r$. To prove the claim, assume by contradiction that r is the first round in which Duplicator has to falsify a clause in $\mathcal{C}_r$. $r > 0$ because initially $V(0) = \emptyset$ and $\square \notin \mathcal{C}_0$ since $\mathsf{regW}(F \vdash) = k$. Also r has to be a round in which Spoiler selects a new variable because in the variable forgeting rounds $\mathcal{C}_r = \mathcal{C}_{r-1}$. Let x be the variable chosen by Spoiler at round r (for the first time). There must be two clauses $A \vee x$ and $B \vee \overline{x}$ in $\mathcal{C}_r$ and $\alpha_{r-1}(A) = \alpha_{r-1}(B) = 0$. But this implies that $A \vee B \in \mathcal{C}_{r-1}$ because

$i)$ $\mathsf{W}(A \vee B) \leq k-1$ since both A and B are falsified by α_{r-1},

[3] In the proof of the implication $1 \Rightarrow 2$ in Theorem 3.2 in [19] it is assumed that putting together a regular resolution proof of $D \vee x$ and one of $E \vee \overline{x}$ one obtains a regular resolution proof of $D \vee E$. This is incorrect since for example, some variable in D could have been resolved in the proof of $E \vee \overline{x}$

$ii)$ there are regular resolution proofs of $A \vee x$ and $B \vee \overline{x}$ of clause width at most $k-1$ and

$iii)$ these proofs do not resolve x since $A \vee x$, $B \vee \overline{x} \in \mathcal{C}_r$.

Putting together the regular resolution proofs of $A \vee x$ and $B \vee \overline{x}$ we would get a regular resolution proof of $A \vee B$. Observe that no variable in $A \vee B$ has been resolved in this proof because $\alpha_{r-1}(A) = \alpha_{r-1}(B) = 0$ which shows that all variables in $A \vee B$ have been selected by Spoiler by round $r-1$. Also $x \notin V(r-1)$ since we are dealing with a regular game. But this would imply that Duplicator would have falsified $A \vee B \in \mathcal{C}_{r-1}$ at the previous round, contradicting the hypothesis stating that r is the first round in which Duplicator falsifies something. □

For our results we need a further version of the SD-game characterizing the width in Davis-Putnam resolution: The *Davis-Putnam SD-game* is played on CNF F in a very similar way as the original SD-game, except for the fact that before the game starts the Spoiler has to give the order σ in which the variables are going to be queried. At each round the Spoiler queries the next *unassigned* variable in σ and Duplicator gives it a value changing its state to *assigned*, or he decides to forget some assigned variable (and its state becomes *queried*). The rest of the game is exactly as in the previous versions. We define $\mathsf{dpsd}(F)$ as the minimum possible value of a terminating Davis-Putnam-SD-game on F.

Theorem 3. *Let F be an unsatisfiable CNF. Then* $\mathsf{dpW}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{dpsd}(F) - 1\}$

Proof. The proof is very similar to that of 2 and we omit some details. To show that for an unsatisfiable CNF, $\mathsf{dpsd}(F) \leq \mathsf{dpW}(F \vdash) + 1$, consider a Davis-Putnam resolution refutation of width k of F. A strategy for Spoiler is to follow the order of variables of this refutation starting at the empty clause in the refutation and move towards a clause in F. In each step, he queries the next variable in σ. If this is the variable being resolved, then depending on Spoiler's answer he moves to the parent clause that is negated by the partial assignment constructed so far. He then deletes from the partial assignment all the variables that are not in the actual clause. If the next variable in the order σ is not the variable being resolved, then independently of Spoiler's assignment he forgets it in the next round. Proceeding this way, a clause in F is negated and Spoiler needs at most $k+1$ variables in memory.

For the other direction, let $0 < k = \mathsf{dpW}(F \vdash)$ and suppose $k > \mathsf{W}(F)$, we give a strategy for Duplicator under which $\mathsf{dpsd}(F) \geq k+1$. Let σ be the ordering of the variables selected by the Spoiler. We use the notation of the previous Theorem. For a round r, let $\mathcal{C}_r$ be the set of all clauses C satisfying:

$i)$ $\mathsf{W}(C) \leq k-1$,

$ii)$ C can be derived from F using Davis-Putnam resolution of clause width at most $k-1$ and

$iii)$ in the DP resolution proof of C no variable in $V(r)$ is resolved.

The strategy for Duplicator is to assign a value for the variable chosen by Spoiler at round r, that does not falsify a clause in $\mathcal{C}_r$. Again this is always possible and the proof for this follows step by step the proof of the same claim in Theorem 2. □

3 A game characterization of resolution variable space

We give a new characterization of resolution variable space in terms of a two player game. This result holds for any CNF formula and it is therefore quite independent of the rest of the paper. We include it here since it will be used to show that the invisible robber game characterizes variable space in Tseitin formulas. The game is a *non-interactive* version of the Spoiler-Duplicator game defined by Atserias and Dalmau [4] in order to characterize resolution width, considered in the previous section.

Given an unsatisfiable formula F in CNF with variable set V, in our two player game, Player 1 constructs step by step a finite list $L = L_0, L_1, \dots, L_k$ of sets of variables, $L_i \subseteq V$. Starting by the empty set, $L_0 = \emptyset$, in each step he can either add variables to the previous set, or delete variables from it. The *cost* of the game is the size of the largest set in the list.

Once Player 1 finishes his list, Player 2 has to construct dynamically a partial assignment for the set of variables in the list. In each step i, the domain of the assignment is the set of variables L_i in the list at this step. She starts giving some value to the first set of variables in the list, L_1, in a way that no clause of F is falsified. If variables are added to the set at any step, she has to extend the previous partial assignment to the new domain in any way, but again, no initial clause can be falsified. If a variable is kept from one set to the next one in the list, its value in the assignment remains. If variables are removed from the set at any step, the new partial assignment is the restriction of the previous one to the new domain.

If Player 2 manages to come to the end of the list without having falsified any clause of F at any point, she wins. Otherwise Player 1 wins.

Define $\mathsf{nisd}(F)$ to be the minimum cost of a winning game for Player 1 on F. We prove that for any unsatisfiable formula F the variable space of F coincides exactly with $\mathsf{nisd}(F)$.

Theorem 4. *Let F be an unsatisfiable formula, then* $\mathsf{nisd}(F) \leq \mathsf{Vs}(F \vdash)$.

Proof. Consider a resolution proof π of F as a list of configurations. The strategy of Player 1 consists in constructing a list L of sets of variables, that in each step i contains the variables present in the i-th configuration. The cost for this list is exactly $\mathsf{Vs}(\pi)$.

We call a list of partial assignments *correct* if it is constructed following the rules of the game and does not falsify any clause in F.

We claim that any correct list of partial assignments of Player 2 that does not falsify any clause in F has to satisfy simultaneously all the clauses at the configurations in each step. If this would not be true, let us consider the first step i in which the constructed partial assignment falsifies some clause in the configuration. At step $i-1$ the assignment α_i constructed by Player 2 does not falsify any initial clauses nor any clauses in the configuration. At step i, α_i falsifies some clause in the configuration, but this has to be a new added clause and it can only be a clause of F, contradicting the fact that Player 2 is constructing a correct list of partial assignments.

The argument is completed by observing that there must be a step in π in which the clauses in the configuration are not simultaneously satisfiable. □

Theorem 5. *Let F be an unsatisfiable formula, then* $\mathsf{Vs}(F \vdash) \leq \mathsf{nisd}(F)$.

Proof. Let L be the list of sets of variables constructed by Player 1, containing at each step i a set L_i of at most $\mathsf{nisd}(F)$ variables. We consider for each step i a set of clauses $\mathcal{C}_i$ containing only the variables in L_i. Initially L_1 is some set of variables and $\mathcal{C}_1$ is the set of all clauses that can be derived by resolution (in any number of steps) from the clauses in F containing only variables in L_1. At any step i, if L_i is constructed by adding some new variables to L_{i-1}, $\mathcal{C}_i$ is defined to be the set of clauses that can be derived from the clauses in $\mathcal{C}_{i-1}$ and the clauses in F containing only variables in L_i. If L_i is constructed by deleting some variables from L_{i-1}, $\mathcal{C}_i$ is defined to be the set of clauses in $\mathcal{C}_{i-1}$ that only have variables in the set L_i. By definition $\mathcal{C}_i$ can be always be constructed from $\mathcal{C}_{i-1}$ by using only resolution steps, deletion or inclusion of clauses in F.

Therefore this list of sets of clauses can be written as a resolution proof. At every step in this proof at most $\mathsf{nisd}(F)$ variables are present.

We claim that if L is a winning strategy for Player 1, then at some point i, $\mathcal{C}_i$ must contain the empty clause. This implies the result because it shows that there is a resolution proof of F using at most variable space $\mathsf{nisd}(F)$.

Let us define at each step i the set A_i of partial assignments for the variables in L_i that satisfy all the clauses in $\mathcal{C}_i$, and the set B_i to be the set of partial assignments for the variables in L_i that do not falsify any initial clause and can be constructed by Player 2 following the rules of the game.

We show by induction on i that at each step, $A_i = B_i$. Since at some point i, Player 2 does not have any correct assignment that does not falsify a clause in F, it follows that $A_i = B_i = \emptyset$, which means that $\mathcal{C}_i$ in unsatisfiable and must contain the empty clause by the definition of $\mathcal{C}_i$ and the completeness of resolution.

Initially, A_1 is the set of partial assignments that satisfy $\mathcal{C}_1$ and these assignments satisfy all clauses in F containing only variables of L_1 and are therefore contained in B_1. Conversely, any assignment in B_1 satisfies any clause in $\mathcal{C}_1$ because these assignments satisfy every clause in F with variables in L_1 and any assignment satisfying two parent clauses satisfies also its resolvent.

If L_i is obtained by deleting some variables in L_{i-1}, the partial assignments in the set A_i satisfying $\mathcal{C}_i$ are exactly the restrictions to a smaller set of variables of the set A_{i-1} of assignments satisfying $\mathcal{C}_{i-1}$. Clearly any restriction to L_i of an assignment in A_{i-1} satisfies $\mathcal{C}_i$. In the other direction this is also true, because if an assignment $\alpha \in A_i$ could not be extended to an assignment satisfying $\mathcal{C}_{i-1}$ then applying the partial assignment α to $\mathcal{C}_{i-1}$ we would have an unsatisfiable set of clauses, from which the empty clause could be derived by resolution. But this means that from $\mathcal{C}_{i-1}$ a clause with variables in L_i falsified by α would be derivable by resolution, contradicting the fact that α satisfies all clauses in $\mathcal{C}_i$. By induction $A_{i-1} = B_{i-1}$, but B_i is by definition the set of partial assignments in B_{i-1} restricted to L_i and coincides with A_i.

If L_i is obtained by adding some variables in L_{i-1}, by the same argument as above, the set A_i of partial assignments satisfying $\mathcal{C}_i$ are the extensions to a larger set of variables of the set of assignments A_{i-1}, that satisfy $\mathcal{C}_i$. By hypothesis, $A_{i-1} = B_{i-1}$ The partial assignments that Player 2 can produce are the extension of the ones in B_{i-1} that do not falsify a clause in F, and therefore satisfy the clauses in $\mathcal{C}_i$. Again we have $A_i = B_i$. □

For our results we need to define a restricted version of the non-interactive Spoiler-Duplicator game, in which in the list L of sets of variables produced by Player 1, once a variable is deleted from the list it cannot be included later in the list. For an unsatisfiable formula F let $g(F)$ be the value of this game played on F. We show that

Lemma 6. $\mathsf{dpVs}(F \vdash) \leq g(F)$.

Proof. The proof follows the same ideas as Theorem 5. Let L be the list of sets of variables constructed by Player 1, containing at each step i a set L_i of at most $g(F)$ variables. By simulating a variable deletion step from L_i to L_{i-1} by several steps deleting just one variable we can suppose w.l.o.g. that in a deletion step, exactly one variable is deleted. We consider for each step i a set of clauses $\mathcal{C}_i$ containing only the variables in L_i. Initially L_1 is some set of variables and $\mathcal{C}_1$ is the set of clauses in F containing only variables in L_1. At any step i, if L_i is constructed by adding some new variables to L_{i-1}, $\mathcal{C}_i$ is defined to be the set of clauses in $\mathcal{C}_{i-1}$ and the clauses in F containing only variables in L_i. If L_i is constructed by deleting a variable x from L_{i-1}, $\mathcal{C}_i$ is defined to be the set of clauses in $\mathcal{C}_{i-1}$ that do not contain x plus all the clauses than can be resolved from the clauses in $\mathcal{C}_{i-1}$ by resolving over variable x. By definition $\mathcal{C}_i$ can always be constructed from $\mathcal{C}_{i-1}$ by using only resolution steps, deletion or inclusion of clauses in F. Moreover, this is a Davis-Putnam resolution following the order in which the variables are being removed from the sets in the list. Therefore this list of sets of clauses can be written as a DP resolution proof. At every step in this proof at most $g(F)$ variables are present.

We claim that if L is a winning strategy for Player 1, then at some point i, $\mathcal{C}_i$ must contain the empty clause. This implies the result because it shows that there is a DP resolution proof of F using at most variable space $g(F)$.

As in Theorem 5, let us define at each step i the set A_i of partial assignments for the variables in L_i that satisfy all the clauses in $\mathcal{C}_i$, and the set B_i to be the set of partial assignments for the variables in L_i that do not falsify any initial clause and can be constructed by Player 2 following the rules of the game. Exactly as in Theorem 5 it can be seen by induction on i that at each step, $A_i = B_i$. Since at some point i, Player 2 does not have any correct assignment that does not falsify a clause in F, it follows that $A_i = B_i = \emptyset$, which means that $\mathcal{C}_i$ in unsatisfiable. By the completeness of Davis-Putnam resolution, there is a refutation of the empty clause from the clauses $\mathcal{C}_i$ using only variable space $|L_i| \leq g(F)$. □

4 Cops and Robber Games

We consider a slight variation of the Cops and Robber game from Seymour and Thomas [15] which they used to characterize exactly the tree-width of a graph. We call our version the *Edge* (Cops and Robber) *Game*.

Initially a robber is placed on a vertex of a connected graph G. She can move arbitrarily fast to any other vertex along the edges. The team of cops, directed by one person, wants to capture her, and can always see where she is. They are placed on *edges* and do not move.

Definition 4. *(Edge Cops-Robber Game) Player 1 takes the role of the cops. At any stage he can place a cop on any unoccupied edge or remove a cop from an edge. The robber (Player 2) can then move to any vertex that is reachable from her actual position over a path without cops. Both teams have at any moment perfect information of the position of the other team. Initially no cop is on the graph. The game finishes when the robber is captured. This happens when the vertex she occupies is completely surrounded by cops.*

The value of the game is the maximum number of edge-cops present on the edges at any point in the game. We define $\mathsf{ec}(G)$ *as the minimum game value in a finishing Edge Game on* G.

The only differences between our Edge Cops-Robber Game and the Cops-Robber game from Seymour and Thomas are that here the cops are placed on the edges, while in [15] they were placed on the vertices and that our game ends with the robber surrounded while in the Seymour-Thomas game a cop must occupy the same vertex as the robber.

4.1 The cops-robber game characterizes width on Tseitin formulas

The edge-cops game played on a connected graph G characterizes exactly the minimum width of a resolution refutation of $\mathsf{T}(G, \varphi)$ for any odd marking φ. In order to show this, we use the Spoiler and Duplicator game from Atserias-Dalmau [4] introduced to characterize resolution width. We prove that $\mathsf{ec}(G) = \mathsf{sd}(\mathsf{T}(G, \varphi))$ where $\mathsf{sd}(\mathsf{T}(G, \varphi))$ denotes the value of the Atserias-Dalmau game played on $\mathsf{T}(G, \varphi)$.

Let us observe how the Spoiler-Duplicator game goes when played on the Tseitin formula $\mathsf{T}(G, \varphi)$. In a finishing game on $\mathsf{T}(G, \varphi)$ Spoiler and Duplicator construct a partial assignment α of the edges. Applying α to the variables of $\mathsf{T}(G, \varphi)$ a new graph G' and marking φ' are produced. Consider a partial truth assignment α of some of the variables. Assigning a variable $e = \{x, y\}$ in α to 0 corresponds to deleting the edge e in the graph, and setting the edge variable to 1 corresponds to deleting the edge from the graph and toggling the values of $\varphi(x)$ and $\varphi(y)$. The formula $\mathsf{T}(G', \varphi')$ resulting after applying α to (G, φ) is still unsatisfiable.

We will call a connected component of G' for which the sum of the markings of its vertices is odd, an *odd component*. Initially G is an odd component under φ. By assigning an edge, an odd component can be divided in at most two smaller components, an odd one and an even one. The only way for Spoiler to end the game is to construct an assignment α that assigns values to all the edges of a vertex, contradicting its marking under α. This falsifies one of the clauses corresponding to the vertex.

Theorem 7. *For any connected graph G and any odd marking φ,* $\mathsf{ec}(G) = \mathsf{sd}(\mathsf{T}(G, \varphi))$.

Proof. In order to compare both games, the team of cops will be identified with the Spoiler and the robber will be identified with the Duplicator. Since the variables in $\mathsf{T}(G, \varphi)$ are the edges of G, the action of Spoiler selecting (forgetting) a variable in the Atserias-Dalmau game will be identified with placing (removing) a cop on that edge.

We show first that $\mathsf{ec}(G) \leq \mathsf{sd}(\mathsf{T}(G, \varphi))$. No matter what the answers of Duplicator are, Spoiler has a way to play in which he spends at most $\mathsf{sd}(\mathsf{T}(G, \varphi))$ points at the Spoiler-Duplicator game on $\mathsf{T}(G, \varphi)$. In order to obtain a value smaller or equal than $\mathsf{sd}(\mathsf{T}(G, \varphi))$ in the Edge Game, the cops just have to imitate Spoiler's strategy on $\mathsf{T}(G, \varphi)$. At the same time, any decision of the robber can be identified with an assignment of Duplicator that captures the position of the robber. This is done by considering a Duplicator assigning values in such a way that there is always a unique odd component which corresponds to the subgraph of G isolated by cops where the robber is. At any step in the Edge Game, the following invariant is kept:

The partial assignment produced in the Spoiler-Duplicator game on $\mathsf{T}(G, \varphi)$ defines a unique odd component corresponding to the component of the robber.

Initially the robber is in some vertex of the graph, which is the unique odd component. If in a step of the Spoiler-Duplicator game the edge selected by Spoiler does not cut the component where the robber is, Player 1 can simulate Duplicator's assignment for this variable in any way and the unique odd component of the robber is kept. He can continue with the next decision of Spoiler. At a step right after the component of the robber is cut by the cops, Player 1 can compute an assignment of Duplicator for the last occupied edge, which would create a labeling that identifies the component with the robber as the unique odd component of the graph. This is always possible. Then Player 1 just needs to continue the imitation of Spoiler's strategy for the assignment produced by Duplicator.

At the end of the game Spoiler falsifies an initial clause, and the vertex corresponding to this clause is the unique odd component under the partial assignment. Therefore the cops will be on the edges of a falsified clause, thus catching the robber on the corresponding vertex.

The proof of $\mathsf{ec}(G) \geq \mathsf{sd}(\mathsf{T}(G, \varphi))$ is very similar. Now we consider that there is a strategy for Player 1 in the Edge Game using at most $\mathsf{ec}(G)$ cops, and we want to extract from it a strategy for the Spoiler. He just needs to select (remove) variables is the same way as the cops are being placed (removed). This time, all through the game we have the following invariant:

The component isolated by cops in which the robber is, is an odd component in the Spoiler-Duplicator game.

When the variable (edge) selected does not cut the component where the robber is, Spoiler does not need to do anything. When the last selected variable cuts the component of the robber, by choosing a value for this variable Duplicator decides which one of the two new components is the odd one. Spoiler figures that the robber has gone to the new odd component and asks the cops what to do next in this situation. When the robber is caught, this will be in an odd component of size 1 which all its edges assigned. This partial assignment falsifies the corresponding clause in $\mathsf{T}(G, \varphi)$. □

Using this result, the width characterization from [4] and the fact that $\mathsf{W}(\mathsf{T}(G, \varphi))$ corresponds to the maximum degree of the graph, $\Delta(G)$, we obtain:

Corollary 8. *For any connected graph G and any odd marking φ,*

$$\mathsf{W}(\mathsf{T}(G,\varphi)\vdash) = \max\{\Delta(G), \mathsf{ec}(G)-1\}.$$

4.2 An invisible robber characterizes variable space on Tseitin formulas

We consider now the version of the edge-cops game in which the robber is invisible. That means that the cops strategy cannot depend on the robber and the cops have to explore the whole graph to catch her. As in the visible version of the game, the robber is caught if all the edges around the vertex in which she is, are occupied by cops. For a graph G let $\mathsf{iec}(G)$ be the minimum number of edge-cops needed to catch an invisible robber in G. Let $\mathsf{T}(G,\varphi)$ be the Tseitin formula corresponding to G. We show that $\mathsf{iec}(G)$ corresponds exactly to $\mathsf{Vs}(\mathsf{T}(G,\varphi))$.

Theorem 9. $\mathsf{Vs}(\mathsf{T}(G,\varphi)) = \mathsf{iec}(G)$.

Proof. (i) $\mathsf{Vs}(\mathsf{T}(G,\varphi)\vdash) \leq \mathsf{iec}(G)$. We use the game characterization of variable space from Section 3. Consider the strategy of the cops. At each step the set of variables constructed by Spoiler corresponds to the set of edges (variables) where the cops are. Now consider any list of partial assignments that Player 2 might construct. Any such assignment can be interpreted as deleting some edges and moving the robber to an odd component in the graph. But the invisible robber is caught at some point no matter what she does, and this corresponds to a falsified initial clause.

(ii) $\mathsf{iec}(G) \leq \mathsf{Vs}(\mathsf{T}(G,\varphi)\vdash)$. Now we have a strategy for Spoiler, and the cops just need to be placed on the edges corresponding to the variables selected by Player 1. If the robber could escape, by constructing a list of partial assignments mimicking the robber moves (that is, each time the cops produce a new cut in the component where the robber is, she sets the value of the last assigned variable to make odd the new component where the robber has moved to), Player 2 never falsifies a clause in $\mathsf{T}(G,\varphi)$. □

4.3 A game characterization of depth on Tseitin formulas

We consider now a version of the game in which the cops have to remain on their edges until the end of the game and cannot be reused.

Definition 5. *For a graph G let $\mathsf{lec}(G)$ be the minimum number of edge-cops needed in order to catch a visible robber on G, in the cops-robber game, with the additional condition that the cops once placed, cannot be removed from the edges until the end of the game (lazy cops).*

Theorem 10. *For any connected graph G and any odd marking φ of G,* $\mathsf{D}(\mathsf{T}(G,\varphi)\vdash) = \mathsf{lec}(G)$.

Proof. (i) $\mathsf{D}(\mathsf{T}(G,\varphi)\vdash) \leq \mathsf{lec}(G)$. Based on the strategy of the cops, we construct a Davis-Putnam resolution proof tree of $\mathsf{T}(G,\varphi)$ in which the variables are resolved in the order (from the empty clause) as the cops are being placed on the edges. Starting at the node in the tree corresponding to the empty clause, in each step when a cop is placed on edge e we consider two parent edges, one labeled by e and the other one by $\overline{e}$. A node in the tree is identified by the partial assignment defined by the path going from the empty clause to this node and falsifying all the literals in the path. Each time the cops produce a cut in G, such an assignment defines two different connected components in G, one with odd marking and one with even marking. At this point we keep the construction of the resolution proof leading to this node by considering the resolution of the component with the odd marking, following the cop strategy for the case in which the robber did go to this component. The resolution is Davis-Putnam since each time a cut is produced the variables (edges) on each side of the cut are disjoint.

(ii) $\mathsf{lec}(G) \leq \mathsf{D}(\mathsf{T}(G, \varphi) \vdash)$. Consider a resolution proof π of $\mathsf{T}(G, \varphi)$. Starting by the empty clause, the cops are placed on the edges corresponding to the variables being resolved. At the same time a partial assignment is being constructed (by the robber) that defines a path in the resolution graph starting at the empty clause and going through the clauses that are negated by the partial assignment. If removing the edges where the cops are produces a cut in G, the cops continue from a node in the resolution proof corresponding to an assignment for the last chosen variable that gives odd value to the component where the robber has moved. At the end a clause in $\mathsf{T}(G, \varphi)$ is falsified, which corresponds to the cops being placed in the edges around the robber. The number of cops needed is at most the resolution depth. □

From the the proof of this result follows that in fact for Tseitin formulas the depth of a DP resolution is minimal.

Corollary 11. *Let $G = (V, E)$ be a simple connected graph and let φ be any odd marking of G. Assume that there exist a resolution refutation of $\mathsf{T}(G, \varphi)$ of depth at most k. Then there exists a Davis-Putnam resolution refutation of $\mathsf{T}(G, \varphi)$ of width at most k.*

5 Davis-Putnam resolution and monotone games

We show in this section that the fact that the games can be played in a monotone way, implies that width and variable space in Davis-Putnam resolution are as good as in general resolution in the context for Tseitin formulas.

We need some further notation. For a set S and $k > 0$, we denote the set of subsets of S of size at most k by S^k.

5.1 The visible robber

We recall the vertex-cops game of [15]. Let $G = (V, E)$ be a simple graph and let $Y \subseteq V$. A *Y-flap* is the vertex set of a connected component in $G \setminus Y$. A position in this game is a pair (Y, Q) where $Y \subseteq V$ and Q is an Y-flap. A game can be considered as a sequence of positions, Y represents the set of vertices with cops on them and Q represents the Y-flap where the robber is. The game starts in position $(\emptyset, V)$. Assume that position (Y_i, Q_i) is reached. The cops choose Y_{i+1} such that either $Y_i \subseteq Y_{i+1}$ or $Y_{i+1} \subseteq Y_i$. Then the robber chooses a Y_{i+1}-flap Q_{i+1} such that $Q_i \subseteq Q_{i+1}$ or $Q_{i+1} \subseteq Q_i$. The cops win when $Q_i \subseteq Y_{i+1}$. We say that a sequence of positions $(Y_0, Q_0), \ldots, (Y_t, Q_t)$ is *monotone* if for all $0 \leq i \leq j \leq k \leq t$, $Y_i \cap Y_k \subseteq Y_j$. The main result of Seymour and Thomas states that if k cops can win the game, they can also win monotonically. We will use this result to prove an analogous statement about our edge-games.

We extend the framework of Seymour and Thomas to talk about edges. Now we have $X \subseteq E$. An X-flap is the edge set of a connected component in $G \setminus X$. A position is a pair (X, R) with $X \subseteq E$ and R an X-flap. Assume that a position (X_i, R_i) is reached. The cops choose X_{i+1} such that either $X_i \subseteq X_{i+1}$ or $X_{i+1} \subseteq X_i$. Then the robber chooses an X_{i+1}-flap R_{i+1} such that either $R_i \subseteq R_{i+1}$ or $R_{i+1} \subseteq R_i$. The cops win when $R_i \subseteq X_{i+1}$, that is, when all the edges adjacent to R_i are in X_{i+1}. Note that under this definition if some X isolates more than one vertex, then we will have multiple empty sets as X-flaps. However if the robber moves to such an X-flap she will immediately lose as in the next round the cops remain where they are and $\emptyset \subseteq X$.

Similarly a sequence of positions $(X_0, R_0), \ldots, (X_t, R_t)$ is monotone if for all $0 \leq i \leq j \leq k \leq t$, $X_i \cap X_k \subseteq X_j$.

Definition 6. *Given a graph $G = (V, E)$ the line graph of G is $L(G) = (V', E')$ defined as follows: for every edge $e \in E$ we put a vertex $w_e \in V'$. We then set*

$$E' = \{\{w_{e_1}, w_{e_2}\} : e_1, e_2 \in E, e_1 \cap e_2 \neq \emptyset\}.$$

For $X \subseteq E$ define $L(X) := \{w_e : e \in X\}$ and for $Y \subseteq V'$ define $L^{-1}(Y) = \{e : w_e \in Y\}$.

Proposition 12. *Let $G = (V, E)$ be a graph and let $X \subseteq E$. It follows that $R \subseteq E$ is an X-flap if and only $L(R)$ is an $L(X)$-flap.*

Proof. It is enough to show that any $e_1, e_2 \in E \setminus X$ are reachable from each other in $G \setminus X$ if and only if w_{e_1} and w_{e_2} are reachable from each other in $L(G) \setminus L(X)$. Let $P = e_1, f_1, \dots, f_t, e_2$ be a path in $G \setminus X$ connecting e_1 and e_2. By construction we have a path $w_{e_1}, w_{f_1}, \dots, w_{f_t}, w_{e_2}$ in $L(G) \setminus L(X)$.

Conversely let $w_{e_1}, w_{f_1}, \dots, w_{f_t}, w_{e_2}$ be a path of minimum length between w_{e_1} and w_{e_2} in $L(G) \setminus L(X)$. It is easy to see that $e_1, f_1, \dots, f_t, e_2$ is a path between e_1 and e_2 in $G \setminus X$. □

Theorem 13. *Assume that there is a strategy for the edge-cops game on G with k cops. Then there exists a strategy for the vertex-cops game in $L(G)$ with k cops.*

Proof. Fix a strategy σ for the edge-cops on G, i.e., for every $X \in E^k$ and every X-flap R, $\sigma(X, R) \in E^k$ which guarantees that the robber will eventually be captured. We will inductively construct a sequence $\{(Y_i, Q_i)\}$ of positions in the vertex game on $L(G)$, where Q_is are the responses of the robber, while keeping a corresponding sequence $\{(X_i, R_i)\}$ for the edge game on G. The vertex game starts in position $(Y_0, Q_0) = (\emptyset, V')$ and the edge game starts in $(X_0, R_0) = (\emptyset, E)$. We have $X_1 = \sigma(X_0, R_0)$. In general we set $Y_i = L(X_i)$ and after the robber has responded with Q_i we define $R_i = L^{-1}(Q_i)$, from which we construct $X_{i+1} = \sigma(X_i, R_i)$ and so on. That R_i is an X_i-flap follows immediately from Proposition 12. To see that this is indeed a winning strategy, note that at some point we reach a position with $R_i \subseteq X_{i+1}$. This happens only when $Q_i \subseteq Y_{i+1}$. □

Theorem 14. *Assume that there is a monotone strategy for the vertex-cops game in $L(G)$ with k cops. Then there exists a monotone strategy with k cops for the edge-cops game in G.*

Proof. We will construct a sequence $\{(X_i, R_i)\}$ of positions in the edge game on G while keeping a corresponding sequence $\{(Y_i, Q_i)\}$ of positions in the vertex game on $L(G)$. Note that R_i will be the response of the robber on G. Let σ be a monotone strategy with k vertex-cops on $L(G)$. We will inductively construct $X_i = \{e : w_e \in Y_i\}$ and after the robber has responded with R_i we define $Q_i = L(R_i)$. Proposition 12 implies that Q_i is a Y_i-flap. Since σ is a winning strategy at some point we reach a position with $Q_i \subseteq Y_{i+1}$. This happens only when $R_i \subseteq X_{i+1}$. The monotonicity of the strategy follows. □

Corollary 15. *Let $G = (V, E)$ be a simple connected graph and let φ be any odd marking of G. Assume that there exist a resolution refutation of $\mathsf{T}(G, \varphi)$ of width at most k. Then there exists a Davis-Putnam resolution refutation of $\mathsf{T}(G, \varphi)$ of width at most k.*

Proof. We have seen in Theorem 3 that the Davis-Putnam version of the Spoiler-Duplicator game in [4], in which Spoiler queries the variables following an order σ set before the game starts, characterizes Davis-Putnam resolution width.

Let $\mathsf{W}(\mathsf{T}(G, \varphi)) = k = sd(G) - 1$[4]. There is a winning a strategy for the edge-game on G with at most k edge-cops which by the above results and the monotonicity of the vertex-game can be translated into a

[4]The case $\mathsf{W}(\mathsf{T}(G, \varphi)) = \Delta(G)$ is similar.

monotone strategy with at most k vertex-cops for $L(G)$. This can be translated back into a monotone strategy for the edge-game on G with at most k cops. By Theorem 7 this implies that the Spoiler-Duplicator game on $\mathsf{T}(G, \varphi)$ brings at most k points. Moreover since the edge-game is monotone, cops cannot be replaced on the same edge twice. When the cops create a cut in the graph, the sets of edges on both sides of the cut at disjoint. This implies that it is possible to define a total order of all the edges in the graph so that for any strategy of the Robber, the cops can be placed following that order and will always catch the Robber. The order in which the Cops are removed from edges, as well as the step in which the Robber is caught depends on the strategy of the Robber, but not the order in which the edges are being occupied. Because of this, Theorem 7 shows that in fact the Davis-Putnam version of the Spoiler-Duplicator game on $\mathsf{T}(G, \varphi)$ brings at most k points. By Theorem 3 this shows that the DP resolution width of $\mathsf{T}(G, \varphi)$ is at most k. □

Theorem 14 implies that Theorem 13 states in fact an if and only if condition. From this and the tree-width characterization in terms of the vertex-cops games from [15] it follows that the number of edge-cops needed to win a game on a graph G characterizes the tree-width of the corresponding line graph.

Corollary 16. *Let $G = (V, E)$ be a simple connected graph, then*

$$\mathsf{ec}(G) = \mathsf{vc}(L(G)) = \mathrm{tree-width}(L(G)) + 1.$$

5.2 The invisible robber

A strategy for the cops is formalized by a sequence $(A_0, Z_0), (A_1, Z_1), \ldots, (A_t, Z_t)$ satisfying the following properties:

1. For $0 \leq i \leq t$, $A_i \subseteq V$ (the set of cleared vertices at the ith step) and $Z_i \subseteq E$ (the set of cops at the ith step).
2. $A_0 = \emptyset$ and $A_t = V$.
3. Either $Z_i \subseteq Z_{i+1}$ and A_{i+1} is the union of A_i and the set of vertices whose incident edges are all contained in Z_{i+1}, or $Z_i \supseteq Z_{i+1}$ and A_{i+1} is the subset of A_i of those vertices which are not connected to any vertex in $V \setminus A_i$ with any paths involving no cops in Z_{i+1}.

In a similar way as we did with the visible robber game, we can reduce the edge-game with an invisible robber to the invisible robber vertex-game of Kirousis and Papadimitriou [12] (we will call this game KP). In their game cops are placed on vertices. An edge is *cleared* if both its endpoints have cops. An edge can be *recontaminated* if it is connected to an uncleared edge passing through no cops. It is shown in [12] that the cops can optimally clear all the edges without occupying any vertex twice.

Theorem 17. *Assume that k cops can win the edge-game capturing an invisible robber on $G = (V, E)$. Then k cops can capture the robber in KP game on $L(G)$.*

Proof. Let $(A_0, Z_0), (A_1, Z_1), \ldots, (A_t, Z_t)$ be a strategy of the cops on G. At every step we put a cop on w_e in $L(G)$ if there is a cop in G on e. We observe that at every step i, every edge $(w_e, w_f) \in E(L(G))$ is cleared for any two edges e and f which are incident to a common cleared vertex in A_i. This clearly holds whenever a vertex in G is cleared for the first time (after possible recontamination), since all its incident edges must have a cop. In general note that there is a cop on every edge in $E(A_i, V \setminus A_i)$. This implies that there is no path free of cops from some edge (w_e, w_f) to $(w_{e'}, w_{f'})$ where e and f meet in A_i and e' and f' meet in $V \setminus A_i$. Therefore if v remains in Z_i and e and f are incident with v, then (w_e, w_f) remains clear. Since at the end we have $A_t = V$, all vertices of $L(G)$ will be cleared eventually. □

Theorem 18. *Assume that k cops can monotonically capture the robber in KP game on $L(G)$. Then k cops can monotonically capture the invisible robber in the edge-game on G.*

Proof. At every step we put a cop on e in G if there is a cop on w_e in $L(G)$. This would clearly satisfy monotonicity. However we need to argue that every vertex is cleared eventually. We claim that at every step, the set of cleared vertices in G are those v for which all $(w_e, w_{e'})$ are cleared in $L(G)$ where e and e' are incident with v. Fix v and let C_v be the clique in $L(G)$ on all vertices w_e where e is incident with v. Observe that the first time all $(w_e, w_{e'})$ are cleared in C_v, we necessarily have a cop on all w_e where e in incident with e. In general in the KP game the first time all edges of a clique in a graph are cleared, we necessarily have cops on all the vertices of the clique. By construction at this point v is surrounded. We need to argue that every time v is surrounded it remains cleared as long as all edges in C_v remain cleared. But by construction all paths without cops from v contain only cleared vertices, and thus the result holds. ☐

Corollary 19. *Let $G = (V, E)$ be a simple connected graph and let φ be any odd marking of G. Assume that there exist a resolution refutation of $\mathsf{T}(G, \varphi)$ of variable space at most k. Then there exists a Davis-Putnam resolution refutation of $\mathsf{T}(G, \varphi)$ of variable space at most k.*

Proof. As noted in Lemma 6, the restricted version of the non-interactive Spoiler-Duplicator game is an upper bound for the Davis-Putnam resolution variable space.

Let $\mathsf{Vs}(\mathsf{T}(G, \varphi)) = k$. There is a winning a strategy for the invisible edge-game on G with at most k edge-cops which by the above results and the monotonicity of the invisible vertex-game can be translated into a monotone strategy with at most k vertex-cops for $L(G)$. This can be translated back into a monotone strategy for the invisible edge-game on G with at most k cops. This implies that the restricted version of the non-interactive Spoiler Duplicator game on $\mathsf{T}(G, \varphi)$ (no variable is used more than once) brings at most k points. By Lemma 6 the Davis-Putnam resolution variable space of $\mathsf{T}(G, \varphi)$ is at most k. ☐

Using now the path-width characterization in terms of the vertex-cops invisible robber games from [8] we obtain that the number of edge-cops needed to win a game on a graph G against an invisible robber characterizes the path-width of the corresponding line graph.

Corollary 20. *Let $G = (V, E)$ be a simple connected graph, then*

$$\mathsf{iec}(G) = \mathsf{ivc}(L(G)) = \mathrm{path} - \mathrm{width}(L(G)) + 1.$$

6 New relations between complexity measures for Tseitin formulas

For any unsatisfiable formula F the following inequalities hold:

$$\mathsf{W}(F \vdash) \leq \mathsf{Vs}(F \vdash) \tag{1}$$
$$\mathsf{Vs}(F \vdash) \leq \mathsf{D}(F \vdash) \tag{2}$$
$$\mathsf{Cs}(F \vdash) \leq \mathsf{D}(F \vdash) + 1 \tag{3}$$
$$\mathsf{Cs}(F \vdash) \geq \mathsf{W}(F \vdash) - \mathsf{W}(F) + 1 \tag{4}$$

Here equation 1 follows by definition, equation 2 is proved in [19], equation 4 is the Atserias-Dalmau [4] width-space inequality and equation 3 follows from the following two observations:

1. Any resolution refutation π can be transformed, doubling subproofs, in a tree-like refutation with the same depth of the original proof π.

2. The clause space of a treelike refutation is at most as large as its depth+1 [9].

In general the relationship between variable space and clause space is not clear. It is also an open problem to know whether variable space and depth are polynomially related (see [14, 19]) and if clause space is polynomially bounded in variable space (see Razborov in [14], Open problems). In this Section we answer this questions in the context of Tseitin formulas. We show in Corollary 23 below that for any Tseitin formula $\mathsf{T}(G,\varphi)$ corresponding to a graph G with n vertices,

$$\mathsf{D}(\mathsf{T}(G,\varphi)\vdash) \leq \mathsf{W}(\mathsf{T}(G,\varphi)\vdash)\log n \tag{5}$$

From this and the inequalities above we obtain the following new relations:

$$\mathsf{D}(\mathsf{T}(G,\varphi)\vdash) \leq \mathsf{Vs}(\mathsf{T}(G,\varphi)\vdash)\log n \tag{6}$$

$$\mathsf{Cs}(\mathsf{T}(G,\varphi)\vdash) \leq \mathsf{Vs}(\mathsf{T}(G,\varphi)\vdash)\log n + 1 \tag{7}$$

$$\mathsf{Vs}(\mathsf{T}(G,\varphi)\vdash) \leq (\mathsf{Cs}(\mathsf{T}(G,\varphi)\vdash) + \Delta(G) - 1)\log n. \tag{8}$$

Where the last equation follows since $\mathsf{W}(\mathsf{T}(G,\varphi)) = \Delta(G)$.

That is, in the context of Tseitin formulas $\mathsf{T}(G,\varphi)$:

1. If G is a graph of bounded degree, the width, depth, variable space and clause space for refuting $\mathsf{T}(G,\varphi)$ differ by at most a $\log n$ factor.

2. For any graph G the clause space of refuting $\mathsf{T}(G,\varphi)$ is bounded above by the a $\log n$ factor of the variable space of refuting $\mathsf{T}(G,\varphi)$.

To prove our results, we need two preliminary lemmas.

Lemma 21. *Let $\mathsf{T}(G,\varphi)$ be a Tseitin formula and π be a width k resolution refutation of $\mathsf{T}(G,\varphi)$. From π it is possible to find in linear time in $|\pi|$ a set W of at most $k+1$ variables such that any assignment of these variables when applied to G in the usual way, defines a graph G' and a labeling φ' in which there is some odd connected component with at most $\lceil \frac{|V|}{2} \rceil$ vertices.*

Proof. We use again the Spoiler and Duplicator game from [4]. A way for Spoiler to pay at most $k+1$ points on the game on $\mathsf{T}(G,\varphi)$ is to use the structure of π starting at the empty clause and query each time the variable that is being resolved at the parent clauses. When Duplicator assigns a value to this variable, Spoiler moves to the parent clause falsified by the partial assignment and deletes from this assignment any variables that do not appear in the parent clause. In this way he always reaches at some point an initial clause, falsifying it and thus winning the game. At any point at most $k+1$ variables have to be assigned. To this strategy of Spoiler, Duplicator can oppose the following strategy: She applies the partial assignment being constructed to the initial graph G producing a subgraph G' and a new labeling φ'. Every time a variable e has to be assigned, if e does not produce a new cut in G' she gives to e an arbitrary value. If e cuts an odd component in G' she assigns e with the value that makes the largest of the two new components an odd component. In case e cuts an even component in two, Duplicator gives to e the value which keeps both components even. Observe that with this strategy there is always a unique odd component. Even when Spoiler releases the value of some assigned variable he cannot create more components, he either keeps the same number of components or connects two of them.

While playing the game on $\mathsf{T}(G,\varphi)$ with these two strategies, both players define a path from the empty clause to an initial one. There must be a first clause K along this path in which the partial assignment constructed in the game at the point t in which K is reached, when applied to G, defines a unique odd component of size at most $\lceil \frac{|V|}{2} \rceil$. This is so because the unique odd component initially has size $|V|$ while at the end has size 1. This partial assignment has size at most $k+1$. Not only the odd component, but any component produced by the partial assignment has size at most $\lceil \frac{|V|}{2} \rceil$. This is because at the point before t the odd component was larger than $\lceil \frac{|V|}{2} \rceil$ and therefore any other component had to be smaller than this. At time t Spoiler chooses a variable that when assigned cuts the odd component in two pieces. Duplicator assigns it in such a way that the largest of these two components is odd and has size at most $\lceil \frac{|V|}{2} \rceil$. Therefore the other new component must have at most this size.

Any other assignment of these variables also produces an odd component of size at most $\lceil \frac{|V|}{2} \rceil$. They correspond to other strategies and they all produce the same cuts and components in the graph, just different labeling of the components. Since the initial formula was unsatisfiable there must always be at least one odd component. In order to find the set W of variables, one just has to move on refutation π simulating Spoiler and Duplicator strategies. This can be done in linear time in the size of π. □

Theorem 22. *There is an algorithm that on input a connected graph $G = (V, E)$ with an odd labeling φ and a resolution refutation π of $\mathsf{T}(G, \varphi)$ with width k, produces a tree-like resolution refutation π' of $\mathsf{T}(G, \varphi)$ of depth $k \log(|V|)$.*

Proof. Let $W = \{e_1, \dots e_{|W|}\}$ be a set of variables producing an odd connected component of size at most $\lceil \frac{|V|}{2} \rceil$, as guaranteed by Lemma 21. We can construct a tree-like resolution of depth $|W|$ of the complete formula F_W with $2^{|W|}$ clauses, each containing all variables in W but with a different sign combination.

By the Lemma, each assignment of the variables, when applied to G produces a subgraph G_i and a labeling φ_i with an odd component with at most $\lceil \frac{|V|}{2} \rceil$ vertices. The problem of finding a tree-like refutation for $\mathsf{T}(G, \varphi)$ has been reduced to finding a tree-like resolution refutation for each of the formulas $\mathsf{T}(G_i, \varphi_i)$. But each of the graphs G_i have an odd component with at most $\lceil \frac{|V|}{2} \rceil$ vertices and the problem is to refute the Tseitin formulas corresponding to these components. After at most $\log(|V|)$ iterations we reach Tseitin formulas with just two vertices that can be refuted by trees of depth one. Since W has width at most $k+1$ literals, in each iteration the refutation trees have depth at most k. Putting everything together we get a tree-like refutation of depth at most $k \log(|V|)$.

□

Corollary 23. *For any graph $G = (V, E)$ and any odd labeling φ,*

$$\mathsf{D}(\mathsf{T}(G, \varphi) \vdash) \leq \mathsf{W}(\mathsf{T}(G, \varphi) \vdash) \log(|V|).$$

Corollary 24. *For any graph $G = (V, E)$ and any odd labeling φ*

$$\mathsf{Cs}(\mathsf{T}(G, \varphi) \vdash) \leq \mathsf{Vs}(\mathsf{T}(G, \varphi) \vdash) \log(|V|) + 1.$$

7 Conclusions and Open Problems

We have shown that the measures of width, depth and variable space in the resolution of Tseitin formulas can be exactly characterized in terms of a graph searching game played on the underlying graph. Our game is a slight modification of the well known cops-robber game from Seymour and Thomas. The main motivation for this characterization is the fact that some results in graph searching can be used to solve questions in proof complexity. Using the monotonicity properties of the Seymour and Thomas game, we have proven that the measures of width and variable space in Davis-Putnam resolution coincide exactly with those of general resolution in the context of Tseitin formulas. Previously it was only known that for Tseitin formulas, regular width was within a constant factor of the width in general resolution [2]. The game characterization also inspired new relations between the three resolution measures on Tseitin formulas and we proved that they are all within a logarithmic factor.

We have also obtained a game characterization of variable space for the resolution of general CNF formulas, as a non-interactive version of the Atserias and Dalmau game [4] for resolution width, as well as versions of the Spoiler-Duplicator game that characterize the measure of Davis-Putnam resolution width.

Still open is whether for Tseitin formulas, regular resolution can also simulate general resolution in terms of size, as asked by Urquhart [18]. Also a game characterization of regular or Davis-Putnam variable space remains open. This might be connected with the question of whether the measures of regular and DP variable space coincide for every unsatisfiable formula (we have shown that this is true for the case of Tseitin formulas). Game characterizations for other resolution measures like size or clause space, either for Tseitin or general formulas, would be a very useful tool in proof complexity.

Acknowledgments

The authors would like to thank Osamu Watanabe and the ELC project were this research was started. We are also grateful to Dimitrios Thilikos and to the anonymous referees for helpful comments. Talebanfard's research was supported by ERC grant FEALORA 339691.

References

[1] I. Adler. Marshals, monotone marshals, and hypertree width. *Journal of Graph Theory* 47, 275–296, 2004. 2

[2] M. Alekhnovich and A. A. Razborov. Satisfiability, branch-width and Tseitin tautologies. *Computational Complexity* 20(4), 649–678, 2011. 1, 19

[3] M. Alekhnovich and E. Ben-Sasson and A. A. Razborov and A. Wigderson. Space Complexity in Propositional Calculus *SIAM J. Comput.* 31(4), 1184–1211, 2002. 4

[4] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *18th IEEE Conference on Computational Complexity* 239–247, 2003. 2, 3, 5, 8, 10, 11, 14, 16, 17, 19, 21

[5] P. Beame, C. Beck and R. Impagliazzo. Time-space trade-offs in resolution: Superpolynomial lower bounds for superlinear space. *SIAM J.Comput.* 49, 4, 1612–1645, 2016. 1

[6] C. Beck, J. Nordström and B. Tang. Some trade-off results for polynomial calculus: extended abstract. Proc. of the 45th ACM Symposium on the Theory of Computing, 813–822 (2013). 1

[7] E. Ben-Sasson and A. Wigderson. Short Proofs are Narrow - Resolution made Simple. Journal of the ACM 48(2): 149–169, 2001. 1

[8] J.A. Ellis, I.H. Sudborough and J.S. Turner. The vertex separation and search number of a graph. *Information and Computation* 113(1) 50–79, 1994. 2, 16

[9] J.L. Esteban and J., Torán. Space bounds for resolution. *Information and Computation* 171(1): 84–97, 2001. 1, 4, 5, 17

[10] F.V. Fomin and D. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science* 399 236–245, 2008. 2

[11] G. Gottlob, N. Leone, F. Scarello. Robbers, marshals and guards: game theoretic and logical characterizations of hypertree width. *Journal of Comput. System Sci.* 66, 775–808, 2003. 2

[12] L. M. Kirousis and C. H. Papadimitriou. Searching and Pebbling. *Theoretical Computer Science* 47(3):205–218, 1986. 15, 22

[13] A.S. LaPaugh. Recontamination does not help to search a graph. Tech. report Electrical Engineering and Comp. Science Dept. Princeton University, 1883. 2, 22

[14] A. Razborov. On space and depth in resolution. *Computational Complexity* 27(3), 511-559, 2018. 2, 17

[15] P.D. Seymour and R. Thomas. Graph searching and a Min-Max theorem of tree-width. *Journal of combinatorial theory* Series B 58, 22-35, 1993. 2, 10, 13, 15, 22

[16] J. Torán, Space and Width in Propositional Resolution. *Computational Complexity Column*, Bulletin of EATCS 83: 86–104, 2004. 5

[17] G.S. Tseitin. On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic, Part 2.*, pages 115–125. Consultants Bureau, 1968. 1

[18] A. Urquhart. Hard examples for resolution. *Journal of the ACM* 34, 209–219, 1987 1, 2, 5, 19

[19] A. Urquhart. The depth of resolution proofs. *Studia Logica* 99, 349–364, 2011 5, 6, 16, 17, 21

8 Appendix: Resuming Tables for Measures and Results

8.1 Characterizations of Resolution Proof Measures

On Resolution proofs of unsatisfiable CNF formulas F, we consider the following measures:

Resolution Proofs Measures		
Measure	**Acronym**	**Reference**
Width	$\mathsf{W}(F \vdash)$	Sec. 2
Regular Width	$\mathsf{regW}(F \vdash)$	Subsec 2.1
Davis-Putnam Width	$\mathsf{dpW}(F \vdash)$	Subsec. 2.1
Depth	$\mathsf{D}(F \vdash)$	Sec 2
Variable Space	$\mathsf{Vs}(F \vdash)$	Sec. 2
Regular Variable Space	$\mathsf{regVs}(F \vdash)$	Def. 2
Davis-Putnam Variable Space	$\mathsf{dpVs}(F \vdash)$	Def. 2

Furthermore we consider the following Spoiler-Duplicator games, played on CNFs formula F:

Spoiler Duplicator Games		
Name	**Cost Acronym**	**Reference**
Standard	$\mathsf{sd}(F)$	[4], Subsec. 2.3
Regular	$\mathsf{rsd}(F)$	[19], Subsec. 2.3
Davis-Putnam	$\mathsf{dpsd}(F)$	Subsec. 2.3
non-interactive	$\mathsf{nisd}(F)$	Sec. 3
non-interactive DP	$g(F)$	Before Lem. 6

And we prove the following characterizations:

Characterizations	
Result	**Reference**
$\mathsf{W}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{sd}(F) - 1\}$	[4], Thm. 1
$\mathsf{regW}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{rsd}(F) - 1\}$	[19], Thm. 2
$\mathsf{dpW}(F \vdash) = \max\{\mathsf{W}(F), \mathsf{dpsd}(F) - 1\}$	Thm. 3
$\mathsf{nisd}(F) = \mathsf{Vs}(F \vdash)$	Thm. 4, 5
$\mathsf{dpVs}(F \vdash) \leq g(F)$	Lem. 6

8.2 Resolution Proof Complexity for Tseitin Formulas in terms of Cops-Robber games

We consider the following Cops-Robber games played on undirected graphs G:

Cops-Robber Games		
Players Definition	**Cost Acronym**	**Reference**
Cops on Nodes, Visible Robber	$\mathsf{vc}(G)$	[15]
Cops on Nodes, Invisible Robber	$\mathsf{ivc}(G)$	[12, 13]
Cops on Edges, Visible Robber	$\mathsf{ec}(G)$	Def. 4
Cops on Edges, Invisible Robber	$\mathsf{iec}(G)$	Subsec. 4.2
Cops Stuck on Edges, Visible Robber	$\mathsf{lec}(G)$	Subsec. 4.3

We prove the following results on the Resolution complexity of refuting $\mathsf{T}(G, \varphi)$:

Results on $\mathsf{T}(G, \varphi)$	
Result	**Reference**
$\mathsf{ec}(G)) = \mathsf{sd}(\mathsf{T}(G, \varphi))$	Thm. 7
$\mathsf{W}(\mathsf{T}(G, \varphi) \vdash) = \max\{\Delta(G), \mathsf{ec}(G) - 1\}$.	Cor. 8
$\mathsf{iec}(G) = \mathsf{Vs}(\mathsf{T}(G, \varphi) \vdash)$	Thm. 9
$\mathsf{lec}(G) = \mathsf{D}(\mathsf{T}(G, \varphi) \vdash)$	Thm 10
$\mathsf{dpW}(\mathsf{T}(G, \varphi) \vdash)$ optimal	Cor. 15
$\mathsf{dpVs}(\mathsf{T}(G, \varphi) \vdash)$ optimal	Cor. 19

And finally we obtain the following result relating edge Cops-Robber games played on undirected graphs G with the Vertex-Cops games played on the Line Graph $L(G)$ of G.

Results on Cops-Robber Games	
Result	**Reference**
$\mathsf{ec}(G) = \mathsf{vc}(L(G))$ (=tree-width$(L(G))$)	Cor. 16
$\mathsf{iec}(G) = \mathsf{ivc}(L(G))$ (=path-width$(L(G))$)	Cor. 20

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il