Erdős–Selfridge Theorem for Nonmonotone CNFs

Md Lutfar Rahman ✉
Amazon, Seattle, WA, USA

Thomas Watson ✉
University of Memphis, TN, USA

Abstract

In an influential paper, Erdős and Selfridge introduced the Maker-Breaker game played on a hypergraph, or equivalently, on a monotone CNF. The players take turns assigning values to variables of their choosing, and Breaker's goal is to satisfy the CNF, while Maker's goal is to falsify it. The Erdős–Selfridge Theorem says that the least number of clauses in any monotone CNF with k literals per clause where Maker has a winning strategy is $\Theta(2^k)$.

We study the analogous question when the CNF is not necessarily monotone. We prove bounds of $\Theta(\sqrt{2}^{\,k})$ when Maker plays last, and $\Omega(1.5^k)$ and $O(r^k)$ when Breaker plays last, where $r = (1+\sqrt{5})/2 \approx 1.618$ is the golden ratio.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics

Keywords and phrases Game, nonmonotone, CNFs

Digital Object Identifier 10.4230/LIPIcs.SWAT.2022.31

Funding This work was supported by NSF grants CCF-1657377 and CCF-1942742.

Acknowledgements We thank anonymous reviewers for their comments.

1 Introduction

In 1973, Erdős and Selfridge published a paper [3] with several fundamental contributions, including:

- Being widely regarded as the genesis of the method of conditional expectations. The subsequent impact of this method on theoretical computer science needs no explanation.
- Introducing the so-called Maker-Breaker game, variants of which have since been studied in numerous papers in the combinatorics literature.

We revisit that seminal work and steer it in a new direction. The main theorem from [3] can be phrased in terms of CNFs (conjunctive normal form boolean formulas) that are monotone (they contain only positive literals). We investigate what happens for general CNFs, which may contain negative literals. We feel that the influence of Erdős–Selfridge and the pervasiveness of CNFs in theoretical computer science justify this question as inherently worthy of attention. Our pursuit of the answer uncovers new techniques and invites the development of further techniques to achieve a full resolution in the future.

In the Maker-Breaker game played on a monotone CNF, the eponymous players take turns assigning boolean values to variables of their choosing. Breaker wins if the CNF gets satisfied, and Maker wins otherwise; there are no draws. Since the CNF is monotone, Breaker might as well assign 1 to every variable she picks, and Maker might as well assign 0 to every variable he picks. In the generalization to nonmonotone CNFs, each player can pick which remaining variable and which bit to assign it during their turn. To distinguish this general game, we rename Breaker as T (for "true") and Maker as F (for "false"). The computational complexity of deciding which player has a winning strategy has been studied in [10, 11, 2, 5, 6, 1, 7, 8, 9].

18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022).
Editors: Artur Czumaj and Qin Xin; Article No. 31; pp. 31:1–31:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A CNF is *k-uniform* when every clause has exactly k literals (corresponding to k distinct variables). The Erdős–Selfridge Theorem answers an extremal question: How few clauses can there be in a k-uniform monotone CNF that Maker can win? It depends a little on which player gets the opening move: 2^k if Breaker plays first, and 2^{k-1} if Maker plays first. The identity of the player with the final move doesn't affect the answer for monotone CNFs. In contrast, "who gets the last laugh" matters a lot for general CNFs:

▶ **Theorem 1** (informal). *If* F *plays last, then the least number of clauses in any k-uniform CNF where* F *has a winning strategy is* $\Theta(\sqrt{2}^{\,k})$.

▶ **Theorem 2** (informal). *If* T *plays last, then the least number of clauses in any k-uniform CNF where* F *has a winning strategy is* $\Omega(1.5^k)$ *and* $O(r^k)$ *where* $r = (1+\sqrt{5})/2 \approx 1.618$.

The most involved proof is the $\Omega(1.5^k)$ lower bound in Theorem 2. We conjecture the correct bound is $\Theta(r^k)$.

2 Results

In the *unordered CNF game*, there is a CNF φ and a set of variables X containing all variables that appear in φ and possibly more. The players T and F alternate turns; each turn consists of picking an unassigned variable from X and picking a value 0 or 1 to assign it.[1] The game ends when all variables are assigned; T wins if φ is satisfied (every clause has a true literal), and F wins if φ is unsatisfied (some clause has all false literals). There are four possible patterns according to "who goes first" and "who goes last." If the same player has the first and last moves, then $|X|$ is odd, and if different players have the first and last moves, then $|X|$ is even.

▶ **Definition 3.** *For $k \geqslant 0$ and $a, b \in \{\mathrm{T}, \mathrm{F}\}$, we let $M_{k,a\cdots b}$ be the minimum number of clauses in φ, over all unordered CNF game instances (φ, X) where φ is k-uniform and* F *has a winning strategy when player a has the first move and player b has the last move.*

▶ **Theorem 1** (formal). $M_{k,\mathrm{T}\cdots\mathrm{F}} = \sqrt{2}^{\,k}$ *for even k, and* $1.5\sqrt{2}^{\,k-1} \leqslant M_{k,\mathrm{T}\cdots\mathrm{F}} \leqslant \sqrt{2}^{\,k+1}$ *for odd k.*

Let Fib_k denote the k^{th} Fibonacci number. It is well-known that $\mathrm{Fib}_k = \Theta(r^k)$ where $r = (1+\sqrt{5})/2 \approx 1.618$.

▶ **Theorem 2** (formal). $1.5^k \leqslant M_{k,\mathrm{T}\cdots\mathrm{T}} \leqslant \mathrm{Fib}_{k+2}$ *for all k.*

▶ **Observation 3.** $M_{k,\mathrm{F}\cdots b} = M_{k-1,\mathrm{T}\cdots b}$ *for all $k \geqslant 1$ and $b \in \{\mathrm{T}, \mathrm{F}\}$.*

Proof. $M_{k,\mathrm{F}\cdots b} \leqslant M_{k-1,\mathrm{T}\cdots b}$: Suppose F wins (φ, X) when T moves first, where φ is $(k-1)$-uniform. Then F wins $(\varphi', X \cup \{x_0\})$ when F moves first, where x_0 is a fresh variable (not already in X) and φ' is the same as φ but with x_0 added to each clause. F's winning strategy is to play $x_0 = 0$ first and then use the winning strategy for (φ, X). Note that φ' is k-uniform and has the same number of clauses as φ.

$M_{k-1,\mathrm{T}\cdots b} \leqslant M_{k,\mathrm{F}\cdots b}$: Suppose F wins (φ, X) when F moves first, where φ is k-uniform. Say the opening move in F's winning strategy is $\ell_i = 1$, where $\ell_i \in \{x_i, \overline{x}_i\}$ is some literal. Obtain φ' from φ by removing each clause containing ℓ_i, removing $\overline{\ell}_i$ from each clause

[1] This game is called "unordered" to contrast it with the related TQBF game, in which the variables must be played in a prescribed order.

containing $\overline{\ell}_i$, and removing an arbitrary literal from each clause containing neither ℓ_i nor $\overline{\ell}_i$. Then F wins $(\varphi', X - \{x_i\})$ when T moves first, and φ' is $(k-1)$-uniform and has at most as many clauses as φ. ◀

▶ **Corollary 4.**
- $M_{k,\mathrm{F}\cdots\mathrm{F}} = \sqrt{2}^{\,k-1}$ *for odd* k*, and* $1.5\sqrt{2}^{\,k-2} \leqslant M_{k,\mathrm{F}\cdots\mathrm{F}} \leqslant \sqrt{2}^{\,k}$ *for even* k*.*
- $1.5^{k-1} \leqslant M_{k,\mathrm{F}\cdots\mathrm{T}} \leqslant \mathrm{Fib}_{k+1}$ *for all* k*.*

(Observation 3 requires $k \geqslant 1$, but the bounds in Corollary 4 also hold for $k = 0$ since $M_{0,a\cdots b} = 1$: F wins a CNF with an empty clause, and T wins a CNF with no clauses.)

3 Upper bounds

In this section, we prove the upper bounds of Theorem 1 and Theorem 2 by giving examples of game instances with few clauses where F wins. In [3], Erdős and Selfridge proved the upper bound for the Maker-Breaker game by showing a k-uniform monotone CNF with 2^k clauses where Maker (F) wins. The basic idea is that F can win on the following formula, which is not a CNF:

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee \cdots \vee (x_{2k-1} \wedge x_{2k})$$

Whenever T plays a variable, F responds by assigning 0 to the paired variable. By the distributive law, this expands to a k-uniform monotone CNF with 2^k clauses. We study nonmonotone CNFs, which may have both positive and negative literals.

3.1 F plays last

▶ **Lemma 5.** $M_{k,\mathrm{T}\cdots\mathrm{F}} \leqslant \sqrt{2}^{\,k}$ *for even* k*.*

Proof. F can win on the following formula, which is not a CNF, with variables $X_k = \{x_1, \ldots, x_k\}$.

$$(x_1 \oplus x_2) \vee (x_3 \oplus x_4) \vee \cdots \vee (x_{k-1} \oplus x_k)$$

Whenever T plays a variable, F responds by playing the paired variable to make them equal. To convert this formula to an equivalent CNF, first replace each $(x_i \oplus x_{i+1})$ with $(x_i \vee x_{i+1}) \wedge (\overline{x}_i \vee \overline{x}_{i+1})$. Then by the distributive law, this expands to a k-uniform CNF φ_k where one clause is

$$((x_1 \vee x_2) \vee (x_3 \vee x_4) \vee \cdots \vee (x_{k-1} \vee x_k))$$

and for $i \in \{1, 3, 5, \ldots, k-1\}$, each clause contains either $(x_i \vee x_{i+1})$ or $(\overline{x}_i \vee \overline{x}_{i+1})$. Therefore φ_k has $2^{k/2} = \sqrt{2}^{\,k}$ clauses: one clause for each $S \subseteq \{1, 3, 5, \ldots, k-1\}$. F wins in (φ_k, X_k). ◀

▶ **Lemma 6.** $M_{k,\mathrm{T}\cdots\mathrm{F}} \leqslant \sqrt{2}^{\,k+1}$ *for odd* k*.*

Proof. Suppose φ_{k-1} is the $(k-1)$-uniform CNF with $\sqrt{2}^{\,k-1}$ clauses from Lemma 5 (since $k-1$ is even). We take two copies of φ_{k-1}, and put a new variable x_k in each clause of one copy, and a new variable x_{k+1} in each clause of the other copy. Call this φ_k. Formally:

$$\varphi_k = \bigwedge_{C \in \varphi_{k-1}} (C \vee x_k) \wedge (C \vee x_{k+1})$$

$$X_k = \{x_1, x_2, \ldots, x_{k+1}\}$$

We argue F wins in (φ_k, X_k). If T plays x_k or x_{k+1}, F responds by assigning 0 to the other one. For other variables, F follows his winning strategy for (φ_{k-1}, X_{k-1}) from Lemma 5. Since φ_{k-1} is a $(k-1)$-uniform CNF with $\sqrt{2}^{\,k-1}$ clauses, φ_k is a k-uniform CNF with $2\sqrt{2}^{\,k-1} = \sqrt{2}^{\,k+1}$ clauses. ◀

3.2 T plays last

Before proving Lemma 7 we draw an intuition. We already know that F wins on

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee \cdots \vee (x_{2k-1} \wedge x_{2k}).$$

Now replace each $(x_i \wedge x_{i+1})$ with $(x_i \wedge (\overline{x}_i \vee x_{i+1}))$, which is equivalent. This does not change the function expressed by the formula, so F still wins this $\mathrm{T} \cdots \mathrm{F}$ game. To turn it into a $\mathrm{T} \cdots \mathrm{T}$ game, we can introduce a dummy variable x_0. Since the game is equivalent to a monotone game, neither player has any incentive to play x_0, so F still wins this $\mathrm{T} \cdots \mathrm{T}$ game [4, Proposition 2.1.6].

If we convert it to a CNF, then by the distributive law it will again have 2^k clauses. But this CNF is not uniform – each clause has at least k literals and at most $2k$ literals. We can do a similar construction that balances the CNF to make it uniform. This intuitively suggests that $\sqrt{2}^{\,k} < M_{k,\mathrm{T}\cdots\mathrm{T}} < 2^k$.

▶ **Lemma 7.** $M_{k,\mathrm{T}\cdots\mathrm{T}} \leqslant \mathrm{Fib}_{k+2}$.

Proof. For every $k \in \{0, 1, 2, \ldots\}$ we recursively define a k-uniform CNF φ_k on variables X_k, where $X_k = \{x_0, x_1, \ldots, x_{2k-2}\}$ if $k > 0$, and $X_0 = \{x_0\}$ (these φ_k, X_k are different than in Subsection 3.1):

- $k = 0$: $\varphi_0 = ()$
- $k = 1$: $\varphi_1 = (x_0) \wedge (\overline{x}_0)$
- $k > 1$: $\varphi_k = \bigwedge_{C \in \varphi_{k-1}} (C \vee x_{2k-3}) \wedge \bigwedge_{C \in \varphi_{k-2}} (C \vee \overline{x}_{2k-3} \vee x_{2k-2})$

Now we argue F wins in (φ_k, X_k). F's strategy is to assign 0 to at least one variable from each pair $\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \ldots, \{x_{2k-3}, x_{2k-2}\}$. Whenever T plays from a pair, F responds by assigning 0 to the other variable. After T plays x_0, F picks a fresh pair $\{x_i, x_{i+1}\}$ where i is odd and assigns one of them 0, then "chases" T until T plays the other from $\{x_i, x_{i+1}\}$. Here the "chase" means whenever T plays from a fresh pair, F responds by assigning 0 to the other variable in that pair. After T returns to $\{x_i, x_{i+1}\}$, then F picks another fresh pair to start another chase, and so on in phases. We prove by induction on k that this strategy ensures φ_k is unsatisfied:

- $k = 0$: φ_0 is obviously unsatisfied.
- $k = 1$: φ_1 is obviously unsatisfied.
- $k > 1$: By induction, both φ_{k-1} and φ_{k-2} are unsatisfied. Now φ_k is unsatisfied since: By F's strategy, at least one of $\{x_{2k-3}, x_{2k-2}\}$ is assigned 0. If $x_{2k-3} = 0$ then one of the clauses of φ_k that came from φ_{k-1} is unsatisfied. If $x_{2k-3} = 1$ and $x_{2k-2} = 0$ then one of the clauses of φ_k that came from φ_{k-2} is unsatisfied.

Letting $|\varphi_k|$ represent the number of clauses in φ_k, we argue $|\varphi_k| = \mathrm{Fib}_{k+2}$ by induction on k:

- $k = 0$: $|\varphi_0| = 1 = \mathrm{Fib}_2$.
- $k = 1$: $|\varphi_1| = 2 = \mathrm{Fib}_3$.
- $k > 1$: By induction, $|\varphi_{k-1}| = \mathrm{Fib}_{k+1}$ and $|\varphi_{k-2}| = \mathrm{Fib}_k$. So

$$|\varphi_k| = |\varphi_{k-1}| + |\varphi_{k-2}| = \mathrm{Fib}_{k+1} + \mathrm{Fib}_k = \mathrm{Fib}_{k+2}.$$

Therefore $M_{k,\mathrm{T}\cdots\mathrm{T}} \leqslant \mathrm{Fib}_{k+2}$. ◀

4 Lower bounds

4.1 Notation

In the proofs, we will define a potential value $p(C)$ for each clause C. The value of $p(C)$ depends on the context. If φ is a CNF (any set of clauses), then the potential of φ is $p(\varphi) = \sum_{C\in\varphi} p(C)$. The potential of a literal ℓ_i with respect to φ is defined as $p(\varphi, \ell_i) = p(\{C \in \varphi : \ell_i \in C\})$. When we have a particular φ in mind, we can abbreviate $p(\varphi, \ell_i)$ as $p(\ell_i)$.

Suppose φ is a CNF and ℓ_i, ℓ_j are two literals. We define the potentials of different sets of clauses based on which of ℓ_i, ℓ_j, and their complements exist in the clause. For example, $a(\varphi, \ell_i, \ell_j)$ is the sum of the potentials of clauses in φ that contain both ℓ_i, ℓ_j.

	ℓ_j	$\overline{\ell}_j$	neither ℓ_j nor $\overline{\ell}_j$
ℓ_i	a	b	c
$\overline{\ell}_i$	d	e	f
neither ℓ_i nor $\overline{\ell}_i$	g	h	

$$\begin{aligned}
a(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \ell_i \in C \text{ and } \ell_j \in C\}) \\
b(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \ell_i \in C \text{ and } \overline{\ell}_j \in C\}) \\
c(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \ell_i \in C \text{ and } \ell_j \notin C \text{ and } \overline{\ell}_j \notin C\}) \\
d(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \overline{\ell}_i \in C \text{ and } \ell_j \in C\}) \\
e(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \overline{\ell}_i \in C \text{ and } \overline{\ell}_j \in C\}) \\
f(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \overline{\ell}_i \in C \text{ and } \ell_j \notin C \text{ and } \overline{\ell}_j \notin C\}) \\
g(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \ell_i \notin C \text{ and } \overline{\ell}_i \notin C \text{ and } \ell_j \in C\}) \\
h(\varphi, \ell_i, \ell_j) &= p(\{C \in \varphi : \ell_i \notin C \text{ and } \overline{\ell}_i \notin C \text{ and } \overline{\ell}_j \in C\})
\end{aligned}$$

We can abbreviate these quantities as a, b, c, d, e, f, g, h in contexts where we have particular φ, ℓ_i, ℓ_j in mind. Also the following relations hold:

$$\begin{aligned}
p(\ell_i) &= a + b + c \\
p(\overline{\ell}_i) &= d + e + f \\
p(\ell_j) &= a + d + g \\
p(\overline{\ell}_j) &= b + e + h
\end{aligned}$$

When we assign $\ell_i = 1$ (i.e., assign $x_i = 1$ if ℓ_i is x_i, or assign $x_i = 0$ if ℓ_i is $\overline{x}_i$), φ becomes the *residual* CNF denoted $\varphi[\ell_i = 1]$ where all clauses containing ℓ_i get removed, and the literal $\overline{\ell}_i$ gets removed from remaining clauses.

4.2 F plays last

▶ **Lemma 8.** *$M_{k,\mathrm{T}\cdots\mathrm{F}} \geqslant \sqrt{2}^{\,k}$ for even k.*

Proof. Consider any T $\cdots$ F game instance (φ, X) where φ is a k-uniform CNF with $< \sqrt{2}^{\,k}$ clauses and $|X|$ is even. We show T has a winning strategy. In this proof, we use $p(C) = 1/\sqrt{2}^{\,|C|}$. A *round* consists of a T move followed by an F move.

▷ Claim 9. In every round, there exists a move for T such that for every response by F, we have $p(\psi) \geqslant p(\psi')$ where ψ is the residual CNF before the round and ψ' is the residual CNF after the round.

At the beginning we have $p(C) = 1/\sqrt{2}^{\,k}$ for each clause $C \in \varphi$, so $p(\varphi) < \sqrt{2}^{\,k}/\sqrt{2}^{\,k} = 1$. By Claim 9, T has a strategy guaranteeing that $p(\psi) \leqslant p(\varphi) < 1$ where ψ is the residual CNF after all variables have been played. If this final ψ contained a clause, the clause would be empty and have potential $1/\sqrt{2}^{\,0} = 1$, which would imply $p(\psi) \geqslant 1$. Thus the final ψ must have no clauses, which means φ got satisfied and T won. This concludes the proof of Lemma 8, except for the proof of Claim 9. ◀

Proof of Claim 9. Let ψ be the residual CNF at the beginning of a round. T picks a literal ℓ_i maximizing $p(\psi, \ell_i)$ and plays $\ell_i = 1$.[2] Suppose F responds by playing $\ell_j = 1$, and let ψ' be the residual CNF after F's move. Letting the a, b, c, d, e, f, g, h notation be with respect to ψ, ℓ_i, ℓ_j, we have

$$p(\psi) - p(\psi') = a + b + c + d + g - \big(e + (\sqrt{2} - 1)(f + h)\big)$$

because:

- Clauses from the a, b, c, d, g groups are satisfied and removed (since they contain $\ell_i = 1$ or $\ell_j = 1$ or both), so their potential gets multiplied by 0.
- Clauses from the e group each shrink by two literals (since they contain $\overline{\ell}_i = 0$ and $\overline{\ell}_j = 0$), so their potential gets multiplied by $\sqrt{2} \cdot \sqrt{2} = 2$.
- Clauses from the f, h groups each shrink by one literal, so their potential gets multiplied by $\sqrt{2}$.

By the choice of ℓ_i, we have $p(\ell_i) \geqslant p(\overline{\ell}_i)$ and $p(\ell_i) \geqslant p(\overline{\ell}_j)$ with respect to ψ, in other words, $a + b + c \geqslant d + e + f$ and $a + b + c \geqslant b + e + h$. Thus $p(\psi) \geqslant p(\psi')$ because

$$\begin{aligned} a + b + c + d + g \;\geqslant\; a + b + c \;\geqslant\; \tfrac{1}{2}(d + e + f) + \tfrac{1}{2}(b + e + h) \;&\geqslant\; e + \tfrac{1}{2}(f + h) \\ &\geqslant\; e + (\sqrt{2} - 1)(f + h). \end{aligned}$$

◁

Note: It did not matter whether k is even or odd! Lemma 8 is true for any k. Lemma 10 actually uses oddness of k. The main idea is to exploit the slack $1/2 \geqslant \sqrt{2} - 1$ that appeared at the end of the proof of Claim 9.

▶ **Lemma 10.** *$M_{k,\mathrm{T}\cdots\mathrm{F}} \geqslant 1.5\sqrt{2}^{\,k-1}$ for odd k.*

Proof. Consider any T $\cdots$ F game instance (φ, X) where φ is a k-uniform CNF with $< 1.5\sqrt{2}^{\,k-1}$ clauses and $|X|$ is even. We show T has a winning strategy. In this proof, we use

$$p(C) = \begin{cases} 1/\sqrt{2}^{\,|C|} & \text{if } |C| \text{ is even.} \\ 1/1.5\sqrt{2}^{\,|C|-1} & \text{if } |C| \text{ is odd.} \end{cases}$$

▷ Claim 11. In every round, there exists a move for T such that for every response by F, we have $p(\psi) \geqslant p(\psi')$ where ψ is the residual CNF before the round and ψ' is the residual CNF after the round.

[2] It is perhaps counterintuitive that T's strategy ignores the effect of clauses that contain $\overline{\ell}_i$, which increase in potential after playing $\ell_i = 1$. A more intuitive strategy would be to pick a literal ℓ_i maximizing $p(\psi, \ell_i) - (\sqrt{2} - 1)p(\psi, \overline{\ell}_i)$, which is the overall decrease in potential from playing $\ell_i = 1$; this strategy also works but is trickier to analyze.

At the beginning we have $p(C) = 1/1.5\sqrt{2}^{\,k-1}$ for each clause $C \in \varphi$ (since $|C| = k$, which is odd), so $p(\varphi) < 1.5\sqrt{2}^{\,k-1}/1.5\sqrt{2}^{\,k-1} = 1$. By Claim 11, T has a strategy guaranteeing that $p(\psi) \leqslant p(\varphi) < 1$ where ψ is the residual CNF after all variables have been played. If this final ψ contained a clause, the clause would be empty and have potential $1/\sqrt{2}^{\,0} = 1$ (since 0 is even), which would imply $p(\psi) \geqslant 1$. Thus the final ψ must have no clauses, which means φ got satisfied and T won. This concludes the proof of Lemma 10, except for the proof of Claim 11. ◀

Proof of Claim 11. Let ψ be the residual CNF at the beginning of a round. T picks a literal ℓ_i maximizing $p(\psi, \ell_i)$ and plays $\ell_i = 1$. Suppose F responds by playing $\ell_j = 1$, and let ψ' be the residual CNF after F's move. Letting the a, b, c, d, e, f, g, h notation be with respect to ψ, ℓ_i, ℓ_j, we have

$$p(\psi) - p(\psi') \geqslant a + b + c + d + g - \big(e + \tfrac{1}{2}(f + h)\big)$$

because:

- Clauses from the a, b, c, d, g groups are satisfied and removed (since they contain $\ell_i = 1$ or $\ell_j = 1$ or both), so their potential gets multiplied by 0.
- Clauses from the e group each shrink by two literals (since they contain $\overline{\ell}_i = 0$ and $\overline{\ell}_j = 0$). Here odd-width clauses remain odd and even-width clauses remain even, so their potential gets multiplied by $\sqrt{2} \cdot \sqrt{2} = 2$.
- Clauses from the f, h groups each shrink by one literal. There are two cases for a clause C in these groups:
  - $|C|$ is even, so $p(C) = 1/\sqrt{2}^{\,|C|}$. After C being shrunk by 1, the new clause C' has potential $p(C') = 1/1.5\sqrt{2}^{\,|C'|-1} = 1/1.5\sqrt{2}^{\,|C|-2}$. So the potential of an even-width clause gets multiplied by $p(C')/p(C) = 4/3$.
  - $|C|$ is odd, so $p(C) = 1/1.5\sqrt{2}^{\,|C|-1}$. After C being shrunk by 1, the new clause C' has potential $p(C') = 1/\sqrt{2}^{\,|C'|} = 1/\sqrt{2}^{\,|C|-1}$. So the potential of an odd-width clause gets multiplied by $p(C')/p(C) = 3/2$.

  So their potential gets multiplied by $\leqslant 3/2$ (since $4/3 \leqslant 3/2$).

By the choice of ℓ_i, we have $p(\ell_i) \geqslant p(\overline{\ell}_i)$ and $p(\ell_i) \geqslant p(\overline{\ell}_j)$ with respect to ψ, in other words, $a + b + c \geqslant d + e + f$ and $a + b + c \geqslant b + e + h$. Thus $p(\psi) \geqslant p(\psi')$ because

$$a + b + c + d + g \;\geqslant\; a + b + c \;\geqslant\; \tfrac{1}{2}(d + e + f) + \tfrac{1}{2}(b + e + h) \;\geqslant\; e + \tfrac{1}{2}(f + h). \qquad \lhd$$

4.3 T plays last

▶ **Lemma 12.** $M_{k,\mathrm{T}\cdots\mathrm{T}} \geqslant 1.5^{\,k}$.

Proof. Consider any $\mathrm{T}\cdots\mathrm{T}$ game instance (φ, X) where φ is a k-uniform CNF with $< 1.5^k$ clauses and $|X|$ is odd. We show T has a winning strategy. In this proof, we use $p(C) = 1/1.5^{|C|}$.

For intuition, how can T take advantage of having the last move? She will look out for certain pairs of literals to "set aside" and wait for F to assign one of them, and then respond by assigning the other one the opposite value. We call such a pair "zugzwang," which means a situation where F's obligation to make a move is a disadvantage for F. Upon finding such a pair, T anticipates that certain clauses will get satisfied later, but other clauses containing those literals might shrink when the zugzwang pair eventually gets played. Thus T can update the CNF to pretend those events have already transpired. The normal gameplay of TF rounds (T plays, then F plays) will sometimes get interrupted by FT rounds of playing previously-designated zugzwang pairs. We define the zugzwang condition so that T's modifications won't increase the potential of the CNF (which is no longer simply a residual version of φ). When there are no remaining zugzwang pairs to set aside, we can exploit this fact – together with T's choice of "best" literal for her normal move – to analyze the potential change in a TF round. This allows the proof to handle a smaller potential function and hence more initial clauses, compared to when F had the last move.

We describe T's winning strategy in (φ, X) as Algorithm 1. In the first line, the algorithm declares and initializes ψ, Y, ζ, Z, which are accessed globally. Here ψ is a CNF (initially the same as φ), and ζ is a set (conjunction) of constraints of the form $(\ell_i \oplus \ell_j)$. We consider $(\ell_i \oplus \ell_j)$, $(\ell_j \oplus \ell_i)$, $(\overline{\ell}_i \oplus \overline{\ell}_j)$, $(\overline{\ell}_j \oplus \overline{\ell}_i)$ to be the same constraint as each other. The algorithm maintains the following three invariants:

(1) Y and Z are disjoint subsets of X, and $Y \cup Z$ is the set of unplayed variables, and Y contains all variables that appear in ψ, and Z is exactly the set of variables that appear in ζ, and $|Z|$ is even.

(2) For every assignment to $Y \cup Z$, if ψ and ζ are satisfied, then φ is also satisfied by the same assignment together with the assignment played by T and F so far to the other variables of X.

(3) $p(\psi) < 1$.

Now we argue how these invariants are maintained at the end of the outer loop in Algorithm 1. Invariant (1) is straightforward to see.

▷ Claim 13. Invariant (2) is maintained.

Proof. Invariant (2) trivially holds at the beginning.

Each iteration of the first inner loop maintains (2): Say ψ and ζ are at the beginning of the iteration, and ψ' and ζ' denote the formulas after the iteration. Assume (2) holds for ψ and ζ. To see that (2) holds for ψ' and ζ', consider any assignment to the unplayed variables. We will argue that if ψ' and ζ' are satisfied, then ψ and ζ are satisfied, which implies (by assumption) that φ is satisfied. So suppose ψ' and ζ' are satisfied. Then ψ is satisfied because each clause containing $\ell_i \vee \ell_j$ or containing $\overline{\ell}_i \vee \overline{\ell}_j$ is satisfied due to $(\ell_i \oplus \ell_j)$ being satisfied in ζ', and each other clause is satisfied since it contains the corresponding clause in ψ' which is satisfied. Also, ζ is satisfied since each of its constraints is also in ζ' which is satisfied.

It is immediate that T's and F's "normal" moves in the outer loop maintain (2), because of the way we update ψ and Y.

Each iteration of the second inner loop maintains (2): If an assignment satisfies ψ' and ζ' (after the iteration) then it also satisfies ψ and ζ (at the beginning of the iteration) since T's move satisfies $(\ell_k \oplus \ell_m)$ – and therefore the assignment satisfies φ. ◁

▷ Claim 14. Invariant (3) is maintained.

Proof. Invariant (3) holds at the beginning by the assumption that φ has $< 1.5^k$ clauses (and each clause has potential $1/1.5^k$).

The first inner loop maintains (3) by the following proposition, which we prove later.

► **Proposition 15.** *If* `FindZugzwang()` *returns* (ℓ_i, ℓ_j)*, then* $p(\psi) \geqslant p(\psi')$ *where* ψ *and* ψ' *are the CNFs before and after the execution of* `TfoundZugzwang()`*.*

The second inner loop does not affect (3). In each outer iteration except the last, T's and F's moves from Y maintain (3) by the following proposition, which we prove later.

Algorithm 1 T's winning strategy in (φ, X).

```
initialize  ψ ← φ;  Y ← X;  ζ ← {};  Z ← {}
while game is not over do
    while FindZugzwang() returns a pair (ℓ_i, ℓ_j) do
        TfoundZugzwang(ℓ_i, ℓ_j)
    TplayNormal()
    while F picks x_k ∈ Z and ℓ_k ∈ {x_k, x̄_k} and assigns ℓ_k = 1 do
        TplayZugzwang(ℓ_k)
    if |Y ∪ Z| = 0 then halt
    FplayNormal()

subroutine FindZugzwang():
    if there exist distinct x_i, x_j ∈ Y and ℓ_i ∈ {x_i, x̄_i} and ℓ_j ∈ {x_j, x̄_j} such that (with
      respect to ψ, ℓ_i, ℓ_j): a + e ⩾ (5/4)(b + d) + (1/2)(c + f + g + h) then return (ℓ_i, ℓ_j)
    return NULL

subroutine TfoundZugzwang(ℓ_i, ℓ_j):
    /* T modifies ψ with the intention to make ℓ_i ≠ ℓ_j by waiting for F to touch
      {x_i, x_j} */
    ζ ← ζ ∪ {(ℓ_i ⊕ ℓ_j)};  Z ← Z ∪ {x_i, x_j};  Y ← Y − {x_i, x_j}
    remove from ψ every clause containing ℓ_i ∨ ℓ_j or containing ℓ̄_i ∨ ℓ̄_j
    remove ℓ_i, ℓ̄_i, ℓ_j, ℓ̄_j from all other clauses of ψ

subroutine TplayZugzwang(ℓ_k):
    /* T makes ℓ_m ≠ ℓ_k */
    T picks x_m ∈ Z and ℓ_m ∈ {x_m, x̄_m} such that (ℓ_k ⊕ ℓ_m) ∈ ζ and assigns ℓ_m = 0
    ζ ← ζ − {(ℓ_k ⊕ ℓ_m)};  Z ← Z − {x_k, x_m}

subroutine TplayNormal():
    T picks x_i ∈ Y and ℓ_i ∈ {x_i, x̄_i} maximizing p(ψ, ℓ_i) − p(ψ, ℓ̄_i) and assigns ℓ_i = 1
    ψ ← ψ[ℓ_i = 1];  Y ← Y − {x_i}

subroutine FplayNormal():
    F picks x_j ∈ Y and ℓ_j ∈ {x_j, x̄_j} and assigns ℓ_j = 1
    ψ ← ψ[ℓ_j = 1];  Y ← Y − {x_j};
```

▶ **Proposition 16.** *If* `FindZugzwang()` *returns* NULL*, then* $p(\psi) \geqslant p(\psi')$ *where* ψ *is the CNF before* `TplayNormal()` *and* ψ' *is the CNF after* `FplayNormal()`*.*

This concludes the proof of Claim 14. ◁

Now we argue why T wins in the last outer iteration. Right before `TplayNormal()`, $|Y|$ must be odd by invariant (1), because an even number of variables have been played so far (since T has the first move) and $|X|$ is odd (since T also has the last move) and $|Z|$ is even. Thus, T always has an available move in `TplayNormal()` since $|Y| > 0$ at this point. When T is about to play the last variable $x_i \in Y$ (possibly followed by some Z moves in the second inner loop), all remaining clauses in ψ have width $\leqslant 1$. There cannot be an empty clause in ψ, because then $p(\psi)$ would be $\geqslant 1/1.5^0 = 1$, contradicting invariant (3). There cannot be more than one clause in ψ, because then $p(\psi)$ would be $\geqslant 2/1.5^1 \geqslant 1$. Thus ψ is either empty (already satisfied) or just (x_i) or just $(\overline{x}_i)$, which T satisfies in one move.

At termination, Y and Z are empty, and ψ and ζ are empty and thus satisfied. By invariant (2), this means φ is satisfied by the gameplay, so T wins.

This concludes the proof of Lemma 12 except Proposition 15 and Proposition 16. ◀

Proof of Proposition 15. Since `FindZugzwang()` returns (ℓ_i, ℓ_j), the following holds with respect to ψ, ℓ_i, ℓ_j:

$$a + e \geqslant \tfrac{5}{4}(b + d) + \tfrac{1}{2}(c + f + g + h) \qquad (\spadesuit)$$

We also have

$$p(\psi) - p(\psi') = a + e - \left(\tfrac{5}{4}(b + d) + \tfrac{1}{2}(c + f + g + h)\right)$$

because:

- Clauses from the a, e groups are removed (since they contain $\ell_i \vee \ell_j$ or $\overline{\ell}_i \vee \overline{\ell}_j$), so their potential gets multiplied by 0. (Intuitively, T considers these clauses satisfied in advance since she will satisfy $(\ell_i \oplus \ell_j)$ later.)
- Clauses from the b, d groups each shrink by two literals (since they contain two of $\ell_i, \overline{\ell}_i, \ell_j, \overline{\ell}_j$ which are removed), so their potential gets multiplied by $1.5 \cdot 1.5 = 9/4$. (Some of these four literals will eventually get assigned 1, but since T cannot predict which ones, she pessimistically assumes they are all 0.)
- Clauses from the c, f, g, h groups each shrink by one literal (since they contain one of $\ell_i, \overline{\ell}_i, \ell_j, \overline{\ell}_j$ which are removed), so their potential gets multiplied by $1.5 = 3/2$.

Since ($\spadesuit$) holds, $p(\psi) \geqslant p(\psi')$. ◀

Proof of Proposition 16. In `TplayNormal()`, T picks the literal ℓ_i maximizing $p(\psi, \ell_i) - p(\psi, \overline{\ell}_i)$ and plays $\ell_i = 1$.[3] In `FplayNormal()`, F plays $\ell_j = 1$. With respect to ψ, ℓ_i, ℓ_j we have

$$p(\psi) - p(\psi') = a + b + c + d + g - \left(\tfrac{5}{4}e + \tfrac{1}{2}(f + h)\right)$$

because:

- Clauses from the a, b, c, d, g groups are satisfied and removed (since they contain $\ell_i = 1$ or $\ell_j = 1$ or both), so their potential gets multiplied by 0.

[3] Some other strategies would also work here, but this one is the simplest to analyze.

- Clauses from the e group each shrink by two literals (since they contain $\bar{\ell}_i = 0$ and $\bar{\ell}_j = 0$), so their potential gets multiplied by $1.5 \cdot 1.5 = 9/4$.
- Clauses from the f, h groups each shrink by one literal, so their potential gets multiplied by $1.5 = 3/2$.

By the choice of ℓ_i (i.e., maximizing $p(\ell_i) - p(\bar{\ell}_i)$), we have:

$$\begin{aligned}
&p(\ell_i) - p(\bar{\ell}_i) \geqslant p(\bar{\ell}_j) - p(\ell_j)\\
&\implies a + b + c - d - e - f \geqslant b + e + h - a - d - g\\
&\implies 2a + 0b + 1c + 0d - 2e - 1f + 1g - 1h \geqslant 0 \qquad (\clubsuit)
\end{aligned}$$

Since `FindZugzwang()` returns NULL, ($\spadesuit$) does not hold in ψ. Thus the following holds:

$$\begin{aligned}
&(a + e) < \tfrac{5}{4}(b + d) + \tfrac{1}{2}(c + f + g + h)\\
&\implies -1a + \tfrac{5}{4}b + \tfrac{1}{2}c + \tfrac{5}{4}d - 1e + \tfrac{1}{2}f + \tfrac{1}{2}g + \tfrac{1}{2}h > 0 \qquad (\blacklozenge)
\end{aligned}$$

Thus $p(\psi) \geqslant p(\psi')$ because the linear combination $\frac{9}{16}(\clubsuit) + \frac{1}{8}(\blacklozenge)$ implies:

$$\begin{aligned}
&\tfrac{9}{16}\big(2a + 0b + 1c + 0d - 2e - 1f + 1g - 1h\big) +\\
&\tfrac{1}{8}\big(-1a + \tfrac{5}{4}b + \tfrac{1}{2}c + \tfrac{5}{4}d - 1e + \tfrac{1}{2}f + \tfrac{1}{2}g + \tfrac{1}{2}h\big) > 0\\
&\implies 1a + \tfrac{5}{32}b + \tfrac{5}{8}c + \tfrac{5}{32}d - \tfrac{5}{4}e - \tfrac{1}{2}f + \tfrac{5}{8}g - \tfrac{1}{2}h > 0\\
&\implies 1a + 1b + 1c + 1d - \tfrac{5}{4}e - \tfrac{1}{2}f + 1g - \tfrac{1}{2}h > 0\\
&\implies a + b + c + d + g - \big(\tfrac{5}{4}e + \tfrac{1}{2}(f + h)\big) > 0 \qquad \blacktriangleleft
\end{aligned}$$

References

1. Lauri Ahlroth and Pekka Orponen. Unordered constraint satisfaction games. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 64–75. Springer, 2012.
2. Jesper Byskov. Maker-Maker and Maker-Breaker games are PSPACE-complete. Technical Report RS-04-14, BRICS, Department of Computer Science, Aarhus University, 2004.
3. Paul Erdős and John Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3), 1973.
4. Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. *Positional Games*. Springer, 2014.
5. Martin Kutz. *The Angel Problem, Positional Games, and Digraph Roots*. PhD thesis, Freie Universität Berlin, 2004. Chapter 2: Weak Positional Games.
6. Martin Kutz. Weak positional games on hypergraphs of rank three. In *Proceedings of the 3rd European Conference on Combinatorics, Graph Theory, and Applications (EuroComb)*, pages 31–36. Discrete Mathematics & Theoretical Computer Science, 2005.
7. Md Lutfar Rahman and Thomas Watson. Complexity of unordered CNF games. *ACM Transactions on Computation Theory*, 12(3):18:1–18:18, 2020.
8. Md Lutfar Rahman and Thomas Watson. Tractable unordered 3-CNF games. In *Proceedings of the 14th Latin American Theoretical Informatics Symposium (LATIN)*, pages 360–372. Springer, 2020.
9. Md Lutfar Rahman and Thomas Watson. 6-uniform Maker-Breaker game is PSPACE-complete. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 57:1–57:15. Schloss Dagstuhl, 2021.
10. Thomas Schaefer. Complexity of decision problems based on finite two-person perfect-information games. In *Proceedings of the 8th Symposium on Theory of Computing (STOC)*, pages 41–49. ACM, 1976.
11. Thomas Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16(2):185–225, 1978.