Tetris is Hard, Even to Approximate

Erik D. Demaine* Susan Hohenberger* David Liben-Nowell*

October 21, 2002

Abstract

In the popular computer game of *Tetris*, the player is given a sequence of tetromino pieces and must pack them into a rectangular gameboard initially occupied by a given configuration of filled squares; any completely filled row of the gameboard is cleared and all pieces above it drop by one row. We prove that in the offline version of Tetris, it is NP-complete to maximize the number of cleared rows, maximize the number of tetrises (quadruples of rows simultaneously filled and cleared), minimize the maximum height of an occupied square, or maximize the number of pieces placed before the game ends. We furthermore show the extreme inapproximability of the first and last of these objectives to within a factor of $p^{1-\varepsilon}$, when given a sequence of p pieces, and the inapproximability of the third objective to within a factor of $2-\varepsilon$, for any $\varepsilon > 0$. Our results hold under several variations on the rules of Tetris, including different models of rotation, limitations on player agility, and restricted piece sets.

*Laboratory for Computer Science; Massachusetts Institute of Technology; 200 Technology Square; Cambridge, MA 02139, USA. Email: {edemaine,srhohen,dln}@theory.lcs.mit.edu.

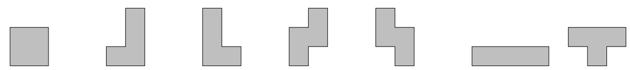


Figure 1: The tetrominoes Sq ("square"), LG ("left gun"), RG ("right gun"), LS ("left snake"), RS ("right snake"), I ("I"), and T ("T").

1 Introduction

Tetris [10] is a popular computer game invented by mathematician Alexey Pazhitnov in the mid-1980s. Tetris is one of the most widespread computer games ever created. By 1988, just a few years after its invention, it was already the best-selling game in the United States and England. Over 50 million copies have been sold worldwide. (Incidentally, Sheff [9] gives a fascinating account of the tangled legal debate over the profits, ownership, and licensing of Tetris.)

In this paper, we embark on the study of the computational complexity of playing Tetris. We consider the *offline* version of Tetris, in which the sequence of pieces that will be dropped is specified in advance. Our main result is a proof that optimally playing offline Tetris is NP-complete, and furthermore is highly inapproximable.

The game of Tetris. Concretely, the game of Tetris is as follows. (We give precise definitions in Section 2, and discuss some variants on these definitions in Section 7.) We are given an initial *gameboard*, which is a rectangular grid with some gridsquares filled and some empty. (In typical Tetris implementations, the gameboard is 20-by-10, and "easy" levels have an initially empty gameboard, while "hard" levels have non-empty initial gameboards, usually with the gridsquares below a certain row filled independently at random.)

A sequence of *tetrominoes*—see Figure 1—is generated, typically probabilistically; the next piece appears in the middle of the top row of the gameboard. The piece falls, and as it falls the player can rotate the piece and slide it horizontally. It stops falling when it lands on a filled gridsquare, though the player has a final opportunity to slide or rotate it before it stops moving permanently. If, when the piece comes to rest, all gridsquares in an entire row h of the game board are filled, row h is *cleared*. All rows above h fall one row lower; the top row of the gameboard is replaced by an entirely unfilled row. As soon as a piece is fixed in place, the next piece appears at the top of the gameboard. To assist the player, typically a one-piece *lookahead* is provided: when the ith piece begins falling, the identity of the $(i + 1)$st piece is revealed.

A player *loses* when a new piece is blocked by filled gridsquares from entirely entering the gameboard. Normally, the player can never win a Tetris game, since pieces continue to be generated until the player loses. Thus the player's objective is to maximize his or her score, which increases as pieces are placed and as rows are cleared.

Our results. We introduce the natural full-information (offline) version of Tetris: we have a *deterministic, finite* piece sequence, and the player knows the identity and order of all pieces that will be presented. (*Games* magazine has, incidentally, posed several Tetris puzzles based on the offline version of the game [7].) We study the offline version because its hardness captures much of the difficulty of playing Tetris; intuitively, it is only easier to play Tetris with complete knowledge of the future, so the difficulty of playing the offline version suggests the difficulty of playing the online version. It also naturally generalizes the one-piece lookahead of implemented versions of Tetris.

It is natural to generalize the Tetris gameboard to m-by-n, since a relatively simple dynamic program solves the case of a constant-size gameboard in time polynomial in the number of pieces. Furthermore, in an attempt to consider the inherent difficulty of the game—and not any accidental difficulty due to the limited reaction time of the player—we begin by allowing the player an arbitrary number of shifts and rotations before the current piece drops in height. (We will restrict these moves to realistic levels later.)

In this paper, we prove that it is NP-complete to optimize any of several natural objective functions for Tetris:

- maximizing the number of rows cleared while playing the given piece sequence;
- maximizing the number of pieces placed before a loss occurs;
- maximizing the number of *tetrises*—the simultaneous clearing of four rows;
- minimizing the height of the highest filled gridsquare over the course of the sequence.

We also prove the extreme inapproximability of the first two (and the most natural) of these objective functions: given an initial gameboard and a sequence of p pieces, for any constant $\varepsilon > 0$, it is NP-hard to approximate to within a factor of $p^{1-\varepsilon}$ the maximum number of pieces that can be placed without a loss, or the maximum number of rows that can be cleared. We also show that it is NP-hard to approximate the minimum height of a filled gridsquare to within a factor of $2 - \varepsilon$.

To prove these results, we first show that the cleared-row maximization problem is NP-hard, and then give various extensions of our reduction for the remaining objectives. Our initial proof of hardness proceeds by a reduction from 3-Partition, in which we are given a set S of $3s$ integers and a bound T, and asked to partition S into s sets of three numbers each, so that the sum of the numbers in each set is exactly T. Intuitively, we define an initial gameboard that forces pieces to be placed into s piles, and give a sequence of pieces so that all of the pieces associated with each integer must be placed into the same pile. The player can clear all rows of the gameboard if and only if all s of these piles have the same height.

A key difficulty in our reduction is that there are only a constant number of piece types, so any interesting component of a desired NP-hard problem instance must be encoded by a sequence of multiple pieces. The bulk of our proof of soundness is devoted to showing that, despite the decoupled nature of a sequence of Tetris pieces, the only way to possibly clear the entire gameboard is to place in a single pile all pieces associated with each integer.

Our reduction is robust to a wide variety of modifications to the rules of the game. In particular, we show that our results also hold in the following settings:

- with restricted player agility—allowing only two rotation/translation moves before each piece drops in height;
- with restricted piece sets—using $\{\mathsf{LG}, \mathsf{LS}, \mathsf{I}, \mathsf{Sq}\}$ or $\{\mathsf{RG}, \mathsf{RS}, \mathsf{I}, \mathsf{Sq}\}$, plus at least one other piece;
- without any losses—i.e., with an infinitely tall gameboard;
- under a wide variety of different rotation models—including the somewhat non-intuitive model that we have observed in real Tetris implementations.

Related work: Tetris. This paper is, to the best of our knowledge, the first consideration of the complexity of playing Tetris. Kostreva and Hartman [8] consider Tetris from a control-theoretic perspective, using dynamic programming to choose the "optimal" move—defined using a heuristic measure of the quality of a configuration. Other previous work has concentrated on the possibility of a *forced eventual loss* (or a *perpetual loss-avoiding strategy*) in the online, infinite version of the game. In other words, under what circumstances can the player be forced to lose, and how quickly? Under what circumstances can the player make the game last for infinitely many moves?

Brzustowski [1] has shown a number of results on forced eventual losses, both positive and negative. He has given a strategy for perpetually avoiding a loss in any (sufficiently large) even-width gameboard using any one-piece pieceset, or any two of the pieces $\{\mathsf{I}, \mathsf{Sq}, \mathsf{RG}, \mathsf{LG}\}$. (The strategies for the piecesets $\{\mathsf{I}, \mathsf{LG}\}$, $\{\mathsf{I}, \mathsf{RG}\}$, and $\{\mathsf{RG}, \mathsf{LG}\}$ rely on the one-piece lookahead.) He has also given such a perpetual loss-avoiding strategy for any (sufficiently large) odd-width gameboard for the piecesets $\{\mathsf{I}\}$, $\{\mathsf{RG}\}$, $\{\mathsf{LG}\}$, and $\{\mathsf{T}\}$.

On the negative side, Brzustowski has shown that perpetual loss-avoidance is impossible for the piecesets $\{\mathsf{RS}\}$, $\{\mathsf{LS}\}$, and $\{\mathsf{Sq}\}$ in odd-width boards. More fundamentally, he has proven that in any size board, if the machine can adversarially choose the next piece (following the lookahead piece) in reaction to the player's moves, then the machine can force an eventual loss using any pieceset that contains $\{\mathsf{LS}, \mathsf{RS}\}$. Burgiel [2] has strengthened this last result for gameboards of width $2n$ for odd n, showing that an alternating sequence of LS's and RS's will eventually cause a loss, regardless of the way in which the player places them. This implies that, if pieces are chosen independently at random with a non-zero probability mass assigned to each of LS and RS, then there is a forced eventual loss with probability one for any such gameboards.

Related work: other games and puzzles. A number of other popular one-player computer games have been shown to be NP-hard, most notably Minesweeper—or, more precisely, the Minesweeper "consistency" problem [6]. See the survey of the first author [3] for a summary of other games and puzzles that have been studied from the perspective of computational complexity. These results form the emerging area of *algorithmic combinatorial game theory*, in which many new results have been established in the past few years.

2 Rules of Tetris

Here we rigorously define the game of Tetris, formalizing the intuition of the previous section. While tedious, we feel that such rigor is necessary so that the many subtle nuances of Tetris become transparent (following immediately from the rules). For concreteness, we have chosen to give very specific rules, but in fact the remainder of this paper is robust to a variety of modifications to these rules; in Section 7, we discuss some variations on these rules for which our results still apply.

In particular, here we consider a particular set of rotation rules—what we call *instantaneous rotation*—that, in our opinion, is the most intuitive and natural model for Tetris rotation. However, we have observed that many Tetris implementations use a different set of rotation rules; in Section 7, we show that our results continue to hold under this observed rotation model.

The gameboard. The *gameboard* is a grid of m rows and n columns, indexed from bottom-to-top and left-to-right. The $\langle i, j \rangle$th *gridsquare* is either *unfilled* (*open*, *unoccupied*) or *filled* (*occupied*). In a legal gameboard, no row is completely filled, and there are no completely empty rows that

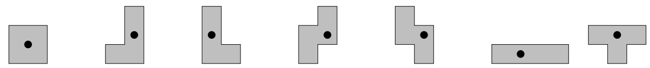


Figure 2: Piece *centers*.

lie below any filled gridsquare. When determining the legality of certain moves, we consider all gridsquares outside the gameboard as always-occupied sentinels.

Game pieces. The seven Tetris pieces, shown in Figure 1, are exactly those connected rectilinear polygons that can be created by assembling four 1-by-1 gridsquares. The *center* of each piece is shown in Figure 2. A *piece state* P is a 4-tuple, consisting of:

1. a *piece type*—$\mathsf{Sq}, \mathsf{LG}, \mathsf{RG}, \mathsf{LS}, \mathsf{RS}, \mathsf{I}$, or T;
2. an *orientation*—$0°, 90°, 180°$, or $270°$—the number of degrees clockwise from the piece's *base orientation* (shown in Figure 1);
3. a *position* of the piece's center on the gameboard, chosen from $\{1, \ldots, m\} \times \{1, \ldots n\}$. The *position* of a Sq is the location of the upper-left gridsquare of the Sq, since its center falls on the boundary of four gridsquares rather than in the interior of one;
4. the value *fixed* or *unfixed*, indicating whether the piece can continue to move.

In an *initial piece state*, the piece is in its base orientation, and the initial position places the highest gridsquares of the piece into row m, and the center into column $\lfloor n/2 \rfloor$, and the piece is unfixed.

Rotating pieces. A *rotation model* is a computable function $R : \langle P, \theta, B \rangle \mapsto P'$, where P and P' are piece states, $\theta \in \{-90°, 90°\}$ is the rotation angle, and B is a gameboard. We impose the following conditions on R:

1. If $P = \langle t, o, \langle i, j \rangle, f \rangle$ and the rotation is *legal*, then $P' = \langle t, (o + \theta) \bmod 360°, \langle i', j' \rangle, f \rangle$ for some i' and j'. If the rotation is *illegal*, then $P' = P$.
2. In determining the legality of a rotation, R only examines an $O(1)$-sized *neighborhood* of the piece P—i.e., only gridsquares within a constant distance of the original position are relevant—and depends only on the neighborhood, and not its location in the gameboard.
3. If all the gridsquares in the neighborhood of P are unfilled, then the rotation is legal.
4. If the rotation is legal, then P' does not occupy any gridsquare already filled in B.

We will impose additional constraints on R in Section 5 to restrict us to *reasonable* rotation models. Intuitively, a reasonable rotation model simply allows for the turning of a piece on the board without any unnatural powers of translation, such as "jumping" to a distant point in the gameboard. Our proof assumes an arbitrary reasonable rotation model; in Section 7, we discuss a number of important reasonable rotation models.

For now, we will consider the *instantaneous rotation model*: fix the piece center (shown in Figure 2), and rotate the piece around that point. The position after rotation is unchanged—i.e., $\langle i', j' \rangle = \langle i, j \rangle$. A rotation is illegal only if it violates Condition 4.

Playing the game. No moves are legal for a piece $P = \langle t, o, \langle i, j\rangle, \mathit{fixed}\rangle$. The following moves are legal for a piece $P = \langle t, o, \langle i, j\rangle, \mathit{unfixed}\rangle$, with current gameboard B:

1. A *clockwise rotation.* The new piece state is $R(P, 90°, B)$.
2. A *counterclockwise rotation.* The new state is $R(P, -90°, B)$.
3. A *slide to the left.* If the gridsquares to the left of the piece are open in B, we can translate P to the left by one column. The new piece state is $\langle t, o, \langle i, j-1\rangle, \mathit{unfixed}\rangle$.
4. A *slide to the right*, similarly. The new piece state is $\langle t, o, \langle i, j+1\rangle, \mathit{unfixed}\rangle$.
5. A *drop* by one row, if all of the gridsquares beneath the piece are open in B. The new piece state is $\langle t, o, \langle i-1, j\rangle, \mathit{unfixed}\rangle$.
6. A *fix*, if at least one gridsquare below the piece is filled in B. The new piece state is $\langle t, o, \langle i, j\rangle, \mathit{fixed}\rangle$.

A *trajectory* σ of a piece P is a sequence of legal moves starting from an initial state and ending with a fix move. The result of a trajectory for a piece P on gameboard B is a new gameboard B', defined as follows:

1. The new gameboard B' is initially B with the gridsquares of P filled.
2. If the piece is fixed so that, for some row r, every gridsquare in row r of B' is full, then row r is *cleared.* For each $r' \geq r$, replace row r' of B' by row $r'+1$ of B'. Row m of B' is an empty row. Multiple rows may be cleared by the fixing of a single piece.
3. If the next piece's initial state is blocked in B', the game ends and the player *loses.*

For a *game* $\langle B_0, P_1, \ldots, P_p\rangle$, a *trajectory sequence* Σ is a sequence $B_0, \sigma_1, B_1, \ldots, \sigma_p, B_p$ so that, for each i, the trajectory σ_i for piece P_i on gameboard B_{i-1} results in gameboard B_i. However, if there is a losing move σ_q for some $q \leq p$ then the sequence Σ terminates at B_q instead of B_p.

The Tetris problem. We will consider a variety of different objectives for Tetris (e.g., maximizing the number of cleared rows, maximizing the number of pieces placed without a loss, etc.) For the decision version of a particular objective Φ, the Tetris problem $\textsc{Tetris}[\Phi]$ is formally as follows:

Given: A Tetris game $\mathcal{G} = \langle B, P_1, P_2, \ldots, P_p\rangle$.

Output: Does there exist a trajectory sequence Σ so that $\Phi(\mathcal{G}, \Sigma)$ holds?

We say that an objective function Φ is *acyclic* when, for all games $\mathcal{G}$, if there is a trajectory sequence Σ so that $\Phi(\mathcal{G}, \Sigma)$ holds, then there is a trajectory sequence Σ' so that $\Phi(\mathcal{G}, \Sigma')$ holds and there are no repeated piece states in Σ'. Most interesting Tetris objective functions are acyclic; in fact, many depend only on the final placement of each piece.

An objective function Φ is *checkable* when, given a game $\mathcal{G}$ and a trajectory sequence Σ, we can compute the truth value of $\Phi(\mathcal{G}, \Sigma)$ in time poly$(|\mathcal{G}|, |\Sigma|)$.

Theorem 2.1 *For any checkable acyclic objective* Φ *we have* $\textsc{Tetris}[\Phi] \in \mathsf{NP}$.

Proof. We are given a Tetris game $\langle B, P_1, \ldots, P_p \rangle$. Here is an NP algorithm for TETRIS[Φ]:

Guess an acyclic trajectory sequence Σ, and confirm that Σ is a legal, acyclic trajectory in time poly($|\Sigma|$). Confirming that all rotations in Σ are legal depends on the computability of the rotation function, and the fact that legality can only depend on the constant-sized neighborhood of the piece.

Since Σ is acyclic, each of its p trajectories can only contain at most $4 \cdot |B| + 1$ states—unfixed once in each position and each orientation, and one final fixed state. Thus $|\Sigma| = \text{poly}(\mathcal{G})$. Since Φ is checkable, we can then in time $\text{poly}(|\mathcal{G}|, |\Sigma|) = \text{poly}(|\mathcal{G}|)$ verify that $\Phi(\mathcal{G}, \Sigma)$ holds, and since Φ is acyclic, guessing an acyclic trajectory sequence Σ suffices. □

The $\Phi(\mathcal{G}, \Sigma)$ that we will initially concern ourselves with is k-cleared-rows($\mathcal{G}, \Sigma$): in the game $\mathcal{G}$, does Σ clear at least k rows without incurring a loss? In Section 7, we will consider a variety of other objective functions.

Lemma 2.2 *The objective* k-cleared-rows *is checkable and acyclic.*

Proof. The objective is acyclic because it only depends on the fixed piece state at the end of each trajectory, so the path in the trajectory is irrelevant; it is checkable since it results from a simple scan of the status of the gameboard after each trajectory in Σ. □

3 The Reduction

In this section, we define a mapping from instances of 3-PARTITION [5, p. 224] to instances of TETRIS[k-cleared-rows]. Recall the 3-PARTITION problem:

Given: A sequence $a_1, \ldots, a_{3s}$ of non-negative integers and a non-negative integer T, so that $T/4 < a_i < T/2$ for all $1 \leq i \leq 3s$ and so that $\sum_{i=1}^{3s} a_i = sT$.

Output: Can $\{a_1, \ldots, a_{3s}\}$ be partitioned into s disjoint subsets $A_1, \ldots, A_s$ so that, for all $1 \leq j \leq s$, we have $\sum_{a_i \in A_j} a_i = T$?

We limit our attention to 3-PARTITION instances that obey the following properties, for technical reasons that will later become apparent:

1. For any set $S \subseteq \{a_1, \ldots, a_{3s}\}$, if $\sum_{a_i \in S} a_i = T$ then $|S| = 3$.
2. T is even.
3. If $\sum_{a_i \in A_j} a_i \neq T$ then $\left| T - \sum_{a_i \in A_j} a_i \right| \geq 3s$.

We can map an arbitrary 3-PARTITION instance into one obeying these properties by multiplying each a_i and T by $4s$. This mapping does not affect whether or not the instance has a valid 3-partition. Property (2) is obviously guaranteed; for property (3), note that, before the multiplication, if $\sum_{a_i \in A_j} a_i \neq T$ then $|T - \sum_{a_i \in A_j} a_i| \geq 1$ since all values are integral, and multiplying by $4s$ multiplies differences by $4s$ as well. For Property (1), note that we still have $T/4 < a_i < T/2$, so it is still the case that if $\sum_{a_i \in S} a_i = T$ then $|S| = 3$.

We choose to reduce from this problem because it is NP-hard to solve 3-PARTITION even if the inputs a_i and T are provided in unary:

Theorem 3.1 (Garey and Johnson [4]) 3-PARTITION *is* NP*-complete in the strong sense.* □

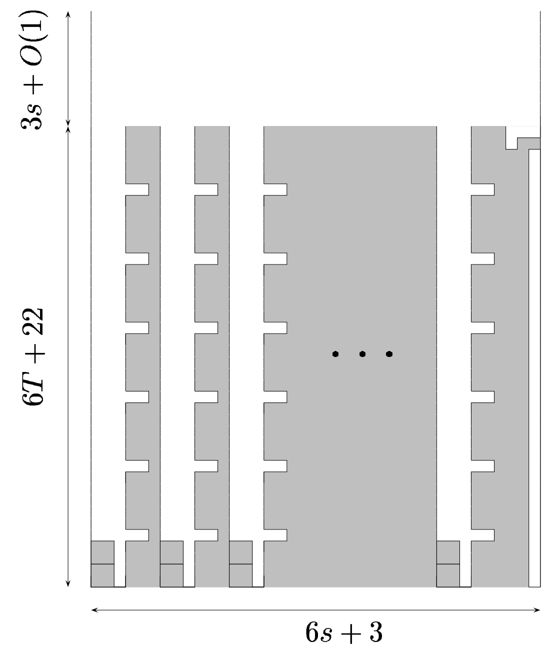


Figure 3: The initial gameboard for a Tetris game mapped from an instance of 3-Partition.

Given a 3-Partition instance $\mathcal{P} = \langle a_1, \ldots, a_{3s}, T \rangle$ obeying the three conditions above, we will produce a Tetris game $\mathcal{G}(\mathcal{P})$ whose gameboard can be completely cleared precisely if $\mathcal{P}$ is a "yes" instance of 3-Partition.

The initial gameboard is shown in Figure 3. Intuitively, there are s *buckets* corresponding to the sets $A_1, \ldots, A_s$ for the 3-Partition problem. The piece sequence will consist of a number of tetrominoes corresponding to each a_i, chosen carefully so that all pieces corresponding to a_i must be placed into the same bucket. There is a legal 3-partition for $\{a_1, \ldots, a_{3s}\}$ exactly when the piles of pieces in each bucket have the same height. The last three columns of the gameboard form a *lock* which prevents any rows from being cleared until the end of the piece sequence; if all buckets are filled exactly to the same height, then the entire board can be cleared using the last portion of our piece sequence.

Formally, our game $\mathcal{G}$ consists of the following:

Initial board: Our gameboard will have $6T+22+(3s+O(1))$ rows and $6s+3$ columns. Intuitively, the factor of six in the height is because each a_i will be represented by a_i+1 blocks of six rows and six columns each; since the set $A_j = \{a_i, a_j, a_k\}$ sums to T, this is $6(T+3) = 6T+18$. In addition to these $6T+18$ rows, there are four rows at the bottom ensuring that the initial blocks are placed correctly.

The top $3s + O(1)$ rows—the $O(1)$ is exactly the size of the neighborhood in the rotation model—are initially empty, and are there solely as a staging area in which to rotate and translate pieces before they fall into the bottom $6T + 22$ rows. We will not mention them again in the construction (and, below, the *highest* row is the $(6T + 22)$nd). Our choice of $3s + O(1)$ as the number of staging rows will be discussed in Section 7.

The remainder of the initial board can be thought of in $s + 1$ logical pieces, the first s of which are six columns wide and the last of which is three columns wide. The first s logical pieces are *buckets*, arranged in the following six-column pattern:

- the first and second columns are empty except that the four lowest rows are full;
- the third column is completely empty;
- the fourth and fifth columns are full in each row $h \not\equiv 5 \pmod{6}$ and empty in each $h \equiv 5 \pmod{6}$;
- the sixth column is completely full;

We call a row $r \equiv 5 \pmod{6}$ a *notch row*, and refer to the unfilled rows of columns 4 and 5 of row r as a *notch*.

The last logical piece is a three-column *lock*, and consists of the following:

- the first column is full except that the highest and second-highest rows are empty;
- the second column is full except that the topmost row is empty;
- the third column is empty except that the second-highest row is full.

Pieces: The sequence of pieces for our game consists of a sequence of pieces for each a_i, followed by a number of additional pieces after all the a_i's. For each integer $a_1, \ldots, a_{3s}$, we have the following pieces:

- the *initiator*, which consists of the sequence $\langle \mathsf{I}, \mathsf{LG}, \mathsf{Sq} \rangle$;
- the *filler*, which consists of the sequence $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ repeated a_i times;
- the *terminator*, which consists of the sequence $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$.

These pieces are given for a_1, a_2, etc., in exactly this order. After the pieces corresponding to a_{3s}, we have the following pieces:

- s successive I's;
- one RG;
- $3T/2 + 5$ successive I's. (Since we enforced that T is even, this is an integral number of pieces.)

Theorem 3.2 *The game $\mathcal{G}(\mathcal{P})$ is polynomial in the size of $\mathcal{P}$.*

Proof. The gameboard has size $6T + 22 + 3s + O(1)$ by $6s + 3$, and the total number of pieces is

$$\sum_{i=1}^{3s} [3 + 5a_i + 2] + s + 1 + \left(\frac{3T}{2} + 5\right) \quad = \quad 16s + 5sT + \frac{3T}{2} + 6.$$

The a_i's and T are represented in unary, so the size of the game is polynomial. □

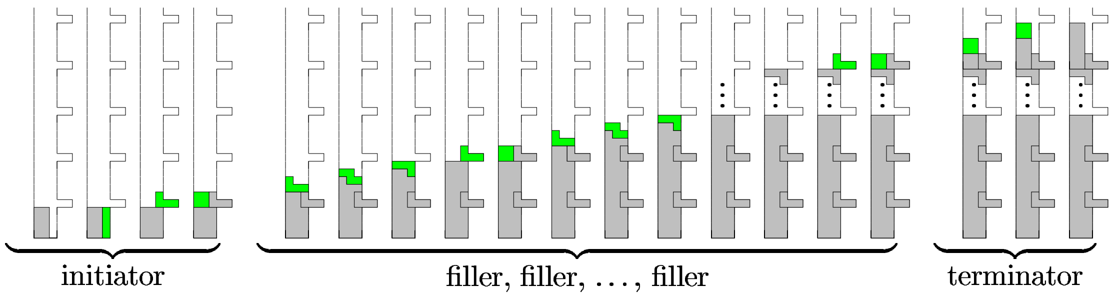


Figure 4: A valid sequence of moves within a bucket.

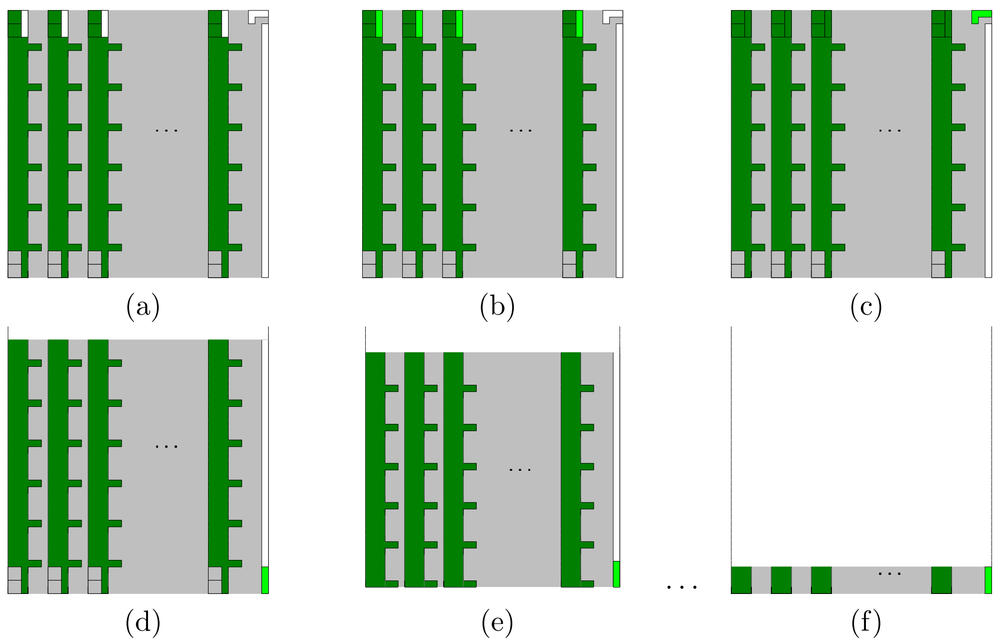


Figure 5: Finishing a valid move sequence.

4 Completeness

Here we show the easier direction of the correctness of our reduction: for a "yes" instance of 3-Partition, we can clear the entire gameboard.

Theorem 4.1 (Completeness) *For any "yes" instance $\mathcal{P}$ of* 3-Partition*, there is a trajectory sequence Σ that clears the entire gameboard of $\mathcal{G}(\mathcal{P})$ without triggering a loss.*

Proof. Since $\mathcal{P}$ is a "yes" instance, there is a partitioning of the a_i's into sets $A_1, \ldots, A_s$ so that $\sum_{a_i \in A_j} a_i = T$. We have ensured that $|A_j| = 3$ for all j. Place all pieces associated with set $A_j = \{x, y, z\}$ into the jth bucket of the gameboard, as illustrated in Figure 4.

Figure 5(a) shows the configuration after all of the pieces associated with $a_1, \ldots, a_{3s}$ have been placed, as follows. After all pieces associated with the number x have been placed into bucket j, the first $4 + 6x + 2$ rows of bucket j are full, and the left-hand two columns of bucket j are filled four rows above that. The pieces associated with the number y fill the next $4 + 6y + 2$ rows, and those with z the next $4 + 6z + 2$ rows, leaving again two columns with four additional rows filled. So the total number of rows filled after the "numbers" is $18+6(x+y+z) = 18+6T$. Doing this for each bucket j yields a configuration in which all buckets are filled up to row $18+6T$, with columns one and two filled up to row $22+6T$.

We next get s successive I's in the sequence. We produce the configuration in Figure 5(b) by dropping one of the I's into the third column of each bucket, to fill rows $19+6T$ through $22+6T$. Now the configuration has the first $22+6T$ rows filled in all of the buckets, and the lock is untouched.

Next we get an RG. Drop it into the slot in the lock, yielding the configuration of Figure 5(c); the first two rows are then cleared. In the resulting configuration, the first $20+6T$ rows of the first $6s+2$ columns are full, and the last column is completely empty.

Figure 5(d–f) shows the final stage of the sequence, as the $3T/2 + 5$ successive I's arrive. Drop each into the last column of the lock. Each of the I's clears four rows; in total, this clears $4 \cdot (3T/2 + 5) = 6T + 20$ rows, clearing the entire the gameboard. The first, second, and last of these I's are illustrated in Figure 5(d–f). □

5 Soundness

Call *valid* any trajectory sequence that clears $6T + 22$ rows in $\mathcal{G}(\mathcal{P})$. We will refer to a move or trajectory as *valid* if it can appear in a valid trajectory sequence. In this section, we show that the existence of a valid strategy for the Tetris game $\mathcal{G}(\mathcal{P})$ implies that $\mathcal{P}$ is a "yes" instance of 3-Partition.

We will often omit reference to $\mathcal{G}(\mathcal{P})$, and refer to its parts simply as the *gameboard* and the *piece sequence*.

5.1 Basic Counting

The soundness of our reduction is fundamentally based upon the observation that, in order to win $\mathcal{G}(\mathcal{P})$, we must fruitfully use every gridsquare in the piece sequence. There are exactly as many unfilled gridsquares in the bottom $6T + 22$ rows of the gameboard as there are gridsquares in the piece sequence; thus, in order to clear the gameboard, we can never place any piece so that it extends beyond the $(6T + 22)$nd row.

Fact 5.1 *The gameboard initially has $64s + 20sT + 6T + 24$ unfilled gridsquares in the bottom $6T + 22$ rows. Of these, there are $64s + 20sT$ unfilled gridsquares in the buckets, and $6T + 24$ unfilled gridsquares in the lock.*

Proof. In each bucket, there are exactly $6T+18$, $6T+18$, $6T+22$, and 0 unfilled gridsquares in the first, second, third, and sixth columns, respectively. The fourth and fifth columns are full in each row $h \not\equiv 5 \pmod 6$ and empty in each $h \equiv 5 \pmod 6$, so the first $6T+18$ rows are exactly one-sixth unfilled; the four highest rows are filled since $6T + 23 \equiv 5 \pmod 6$. Thus there are exactly $T + 3$ unfilled gridsquares in the fourth and fifth columns, and, in total, $20T+64$ unfilled gridsquares per bucket. Therefore, buckets account for exactly $64s + 20sT$ unfilled gridsquares.

In the lock, the first column has exactly two unfilled gridsquares, the second column has exactly one unfilled gridsquare, and the third column has exactly $6T+21$ unfilled gridsquares. Thus there are $2 + 1 + 6T + 21 = 6T + 24$ total in the lock.

Overall, then we have $64s + 20sT + 6T + 24$ unfilled gridsquares. □

Fact 5.2 *The total number of gridsquares in the piece sequence is exactly* $64s + 20sT + 6T + 24$. *Of these, there are* $64s + 20sT$ *gridsquares in sequence of pieces up to (but not including) the* RG*, and* $6T + 24$ *gridsquares in the pieces starting at (and including) the* RG.

Proof. As we calculated in the proof of Lemma 3.2, there are $16s + 5sT + \frac{3T}{2} + 6$ total pieces in the sequence, of which the last $16s + 5sT$ precede the RG. Each covers exactly four gridsquares, so the total number of gridsquares in the piece sequence is $64s + 20sT + 6T + 24$, of which $64s + 20sT$ precede the RG. □

Corollary 5.3 *Any move that places a filled gridsquare above row* $6T + 22$ *is invalid.*

Proof. The total number of filled gridsquares in the input sequence and initial gameboard combined is exactly the number of gridsquares in $6T + 22$ rows. Thus to clear $6T + 22$ rows, every filled gridsquare must be in a cleared row. In particular, we must clear the bottom $6T + 22$ rows (since each of these rows has at least one filled gridsquare initially), and cannot place a filled gridsquare above the $(6T + 22)$nd row. □

5.2 Invalid Moves and Configurations

In this section, we describe general types of configurations from which the game can never be won. In Section 5.4, we will use these results to show that the only possible valid strategy is that of Section 4.

5.2.1 Security of the Lock

Lemma 5.4 *In any valid strategy, none of the pieces* {LG, I, LS, Sq} *can be the first piece placed in a lock column.*

Proof. It is easy to verify that none of {LG, I, LS, Sq} can be placed in such a way to fill even one of the unfilled gridsquares in the lock without also filling a gridsquare beyond the $(6T + 22)$nd row. Therefore, by Corollary 5.3, such a move is invalid. □

There are two important corollaries of this simple lemma:

Corollary 5.5 *In any valid strategy, no rows are cleared before the* RG.

Proof. No rows can be cleared until at least one piece enters the lock columns; by Lemma 5.4, the first piece to do so must be the RG. □

Corollary 5.6 *In any valid strategy:*

1. *all gridsquares of all pieces preceding the* RG *must all be placed into buckets, filling all empty bucket gridsquares.*
2. *all gridsquares of all pieces starting with (and including) the* RG *must be placed into the lock columns, filling all empty lock gridsquares.*

Proof. Immediate from Lemma 5.4, Corollary 5.3, and Facts 5.1 and 5.2. □

In the remainder of this section, we will only consider move sequences that obey these corollaries.

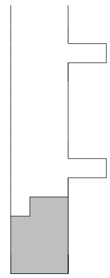


Figure 6: An unapproachable bucket.

5.2.2 Definition of Unfillable Buckets

Call a bucket *unfillable* if it cannot be filled completely using arbitrarily many pieces from the set $\{\mathsf{LG}, \mathsf{LS}, \mathsf{Sq}, \mathsf{I}\}$.

Lemma 5.7 *In any valid strategy, no configuration with an unfillable bucket arises.*

Proof. By Corollary 5.6, all gridsquares of pieces preceding RG must go into buckets, and Facts 5.1 and 5.2 imply that there are exactly the same number of unfilled bucket gridsquares as pre-RG gridsquares in the sequence. If we do not completely fill each bucket, then at least one of these gridsquares will not go into a bucket, violating Corollary 5.6. (Since no rows are cleared, an unfillable configuration can never be made fillable again, and therefore makes the trajectory sequence invalid.) □

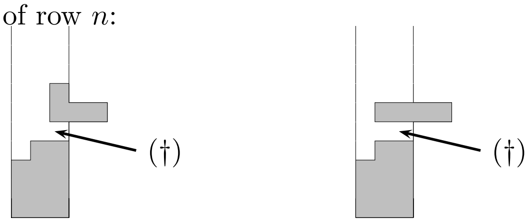
Here we outline a collection of buckets which, if they arise in play, prevent $\mathcal{G}(\mathcal{P})$ from being won. See Figures 6 and 7.

Unapproachable Buckets. In Figure 6, we show an *unapproachable* bucket. This bucket, which can arise during play, is unfillable; in fact, even without a notion of gravity and piece movements, it is impossible even to *tile* such a bucket completely using $\{\mathsf{I}, \mathsf{LG}, \mathsf{LS}, \mathsf{Sq}\}$.

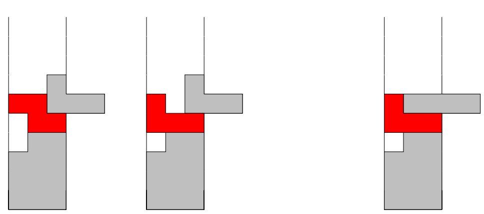
Lemma 5.8 *An unapproachable bucket is unfillable.*

Proof. Suppose not, and consider a filling of a bucket that was unapproachable. Let row n be the first notch row above the unapproachable part of the bucket.

The notch in row n must have been filled by either a LG or an I, either of which also fills the gridsquare in the third column of row n:



However, the gridsquare denoted by (†) can only be filled by a LG or LS (and an LS only in the first case), and either one creates an region that cannot be filled by any Tetris piece:



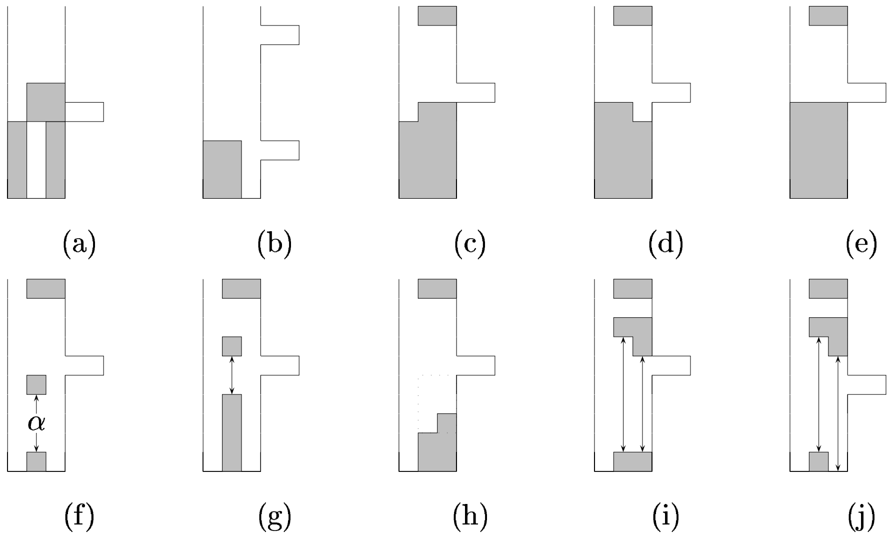


Figure 7: Some unfillable buckets: (a) a hole, (b) a spurned notch, (c) a balconied 011-floored notch, (d) a balconied 110-floored notch, (e) a balconied 111-floored notch, (f) a balconied ($\alpha \not\equiv_4 0$)-veranda in non-notch rows, (g) a balconied 2-veranda, (h) a balconied unfilled sub-notch rectangle, (i) a balconied flat-bottomed 2/3-ceiling discrepancy, (j) a balconied notch-spanning 2/3-ceiling discrepancy.

Thus there was no such filling of the bucket, and unapproachable buckets are unfillable. □

Holes. A *hole* is an unfilled gridsquare at height h in some bucket so that there is a contiguous series of filled gridsquares separating that gridsquare from the empty rows above the buckets. (No piece can ever fill the hole.)

Spurned notches. A *spurned notch* is a bucket in which, for some notch row r, (1) the two gridsquares in the notch are not filled, and (2) the gridsquare in the second column of row r is filled. (The filled gridsquare in the second column prevents any piece from entering the notch.)

Balconied buckets. A *balcony* is a bucket in which, for some row r, two of the three gridsquares in row r are filled. Intuitively, the balcony prevents any piece other than I from "getting past" row r, which means that any unfilled gridsquares below r must be filled entirely using I's. Thus any bucket with a balcony lying over an area that cannot be filled entirely by I's is unfillable.

Call a *α-veranda* an α-long maximal run of consecutive unfilled gridsquares in a column.

We claim that none of the following buckets can be filled using only I's if there is a balcony in row r:

1. a *balconied 111-floored notch*, *011-floored notch*, or *110-floored notch*.

   For some row $h < r$ where h contains an unfilled notch, the three gridsquares in row $h-1$ are all filled (for 111-floored notch), or the second and third are filled (for 011-floored notch), or the first and second are filled (110-floored notch).

The first two are unfillable because an I cannot rotate into the notch; the last is because filling the notch creates a hole, and finishing row $h-1$ creates a 111-floored notch (or a hole, if filling the third column of row $h-1$ simultaneously fills the third column of row h).

2. a *balconied* ($\alpha \not\equiv_4 0$)*-veranda in non-notch rows* or *balconied* 2*-veranda*.

   For some row $h < r$, we have (1) h is the highest unfilled gridsquare in a α-veranda which spans no unfilled notch rows (i.e., only non-notch rows and notch rows in which the notch is already filled), and $\alpha \not\equiv 0 \pmod 4$, or (2) there is a 2-veranda in the hth and $(h+1)$st rows.

   No α-veranda for $\alpha \not\equiv 0 \pmod 4$ can be filled by any number of vertical I's, and a horizontal I will only fit in a notch row. For the balconied ($\alpha \not\equiv_4 0$)-veranda in non-notch rows, there are no relevant notch rows; for a 2-veranda that spans a notch row, filling that notch can only leave a balconied 1-veranda in non-notch rows.

3. a *balconied unfilled sub-notch rectangle*.

   There is an unfilled notch at height $h < r$, and, for column $i \in \{2, 3\}$, and some $1 \leq j \leq 3$, we have the following in column i: the gridsquares at heights $h, h-1, \ldots, h-j$ are unfilled, and the gridsquare at height $h-j-1$ is filled.

   The notch cannot be filled without creating an balconied α-veranda in non-notch rows, for $1 \leq \alpha \leq 3$, and any vertically-placed I in the second or third columns creates a spurned notch or a hole.

4. a *balconied flat-bottomed 2/3-ceiling discrepancy* or a *notch-spanning 2/3-ceiling discrepancy*.

   In a balconied flat-bottomed 2/3-ceiling discrepancy, for some $\alpha \geq 1$, some $1 \leq \beta \leq 3$, and some row $h < r$, we have the following: in both columns two and three, the gridsquares between rows h and $h + \alpha$ inclusive are empty, and the gridsquares in rows $h-1$ are full. In column two (respectively, column three), rows $h+\alpha+1, \ldots, h+\alpha+\beta$ are also empty, and the gridsquare in row $h+\alpha+\beta+1$ is full. In column three (respectively, column two), the gridsquare in row $h+\alpha+1$ is full.

   A balconied notch-spanning 2/3-ceiling discrepancy is just like the above, with the following exceptions: (1) we eliminate the requirement that the second and third gridsquares of row $h-1$ be full, and (2) we add the requirement that there be an unfilled notch in rows $h, \ldots, h+\alpha$.

   For a balconied flat-bottomed 2/3-ceiling discrepancy, since $\beta \not\equiv 0 \pmod 4$, the empty gridsquares in columns two and three cannot both be filled by vertically-oriented I's, and any horizontally-oriented I's placed into notches in these rows fill one gridsquare in both columns, and thus do not resolve this discrepancy. For a balconied notch-spanning 2/3-ceiling discrepancy, we must fill the notch with a horizontal I; once we have done so, the result is a balconied flat-bottomed 2/3-ceiling discrepancy.

These arguments are formalized in Appendix B.

5.3 Reasonable Rotation Models

Which moves are legal, and therefore which instances have valid trajectory sequences, depends on certain properties of the rotation model. So far our only mention of the details of our rotation model has been in the intuition of Section 5.2.2; here we formalize the dependence on the details of the model. Call a rotation model *reasonable* if it satisfies the following four conditions:

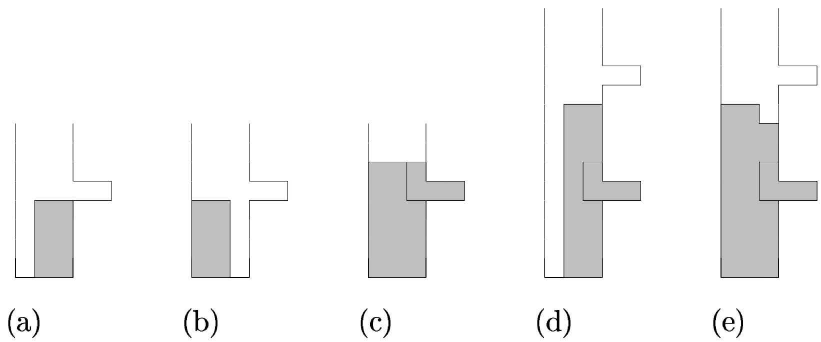


Figure 8: The possibly valid buckets: (a) and (b) unprepped, (c) overflat, (d) tall-plateau, (e) trigger-happy.

1. A piece cannot "jump" from one bucket to another, or into a disconnected region.
2. An LG cannot enter a notch if the gridsquare in the second column of the notch row is filled.
3. For any balcony at height h with the ith column empty in the balcony, the only gridsquares at height $h' \leq h$ that can be filled by any piece other than an I are those in the ith column in rows h and $h-1$.
4. An I cannot enter a notch if the gridsquares in the second and third columns of the row immediately beneath the notch row are filled.

These four conditions are the only ones that we rely on, so this is the only place where we depend on the details of the rotation model. Thus, to show that our reduction holds for any particular rotation model, it suffices to prove that the model is reasonable.

Lemma 5.9 *In any reasonable rotation model, a configuration with a bucket containing any of the following is unfillable: (1) a hole, (2) a spurned notch, (3) a balconied 110-floored notch, 011-floored notch, or 110-floored notch, (4) a balconied unfilled sub-notch rectangle, (5) a balconied* ($\alpha \not\equiv_4 0$)*-veranda in non-notch rows, (6) a balconied* 2*-veranda, (7) a balconied flat-bottomed 2/3-ceiling discrepancy, and (8) a balconied notch-spanning 2/3-ceiling discrepancy.* □

Lemma 5.10 *The instantaneous rotation model of Section 2 is reasonable.* □

We prove Lemmas 5.9 and 5.10 in Appendices B and A, respectively. The proof of Lemma 5.9 formalizes the intuition of the previous section, explaining why we cannot completely fill each of these buckets. The proof of Lemma 5.10 simply checks the conditions of reasonability.

5.4 The Only Way to Play ...

Armed with the results from the previous section, we will show that the only valid moves are those in which we play correctly according to the sequence described in Section 4. We assume (by Corollary 5.4) that no pieces are placed into the lock columns until the RG.

In Figure 8, we give a collection of buckets which can arise during play. We call *unprepped* a bucket, as in (a) or (b), which is filled up to the base of a notch except the top four rows of exactly one of the first and third columns are unfilled. A bucket is *overflat*, as in (c), if it is exactly filled up the row above the top of a notch. A bucket is a *tall-plateau* as in (d), if it has only the first column unfilled in the nine rows starting one row beneath a notch. Finally, a bucket is *trigger-happy* if it is unfilled in exactly the row beneath a notch row plus the third column of row below that.

Call a configuration *unprepped* if each of its buckets is unprepped. Call a configuration *one-x* (respectively, *one-x-one-y*) if exactly one of its buckets is of type x (respectively, one each of types x and y) and the rest are unprepped.

Lemma 5.11 (Initiator Soundness) *In an unprepped configuration, the only possibly valid strategy for $\langle \mathsf{I}, \mathsf{LG}, \mathsf{Sq} \rangle$ is to place all three pieces in some bucket to produce an overflat bucket, yielding a one-overflat configuration.* □

Lemma 5.12 (Filler Soundness) *For the sequence $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$:*

1. *In a one-overflat configuration, the only possibly valid strategy is either (1) to place all pieces in the overflat bucket to produce an overflat bucket, yielding a one-overflat configuration, or (2) to place $\langle \mathsf{LG}, \mathsf{LS} \rangle$ into the overflat bucket and $\langle \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ in an unprepped bucket, yielding a one-tall-plateau-one-trigger-happy configuration.*

2. *In a one-tall-plateau-one-trigger-happy configuration, there is no valid strategy.* □

Lemma 5.13 (Terminator Soundness) *For the sequence $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$,*

1. *In a one-overflat configuration, the only possibly valid strategy is to place both pieces in the overflat bucket to produce an unprepped bucket, yielding an unprepped configuration.*

2. *In a one-tall-plateau-one-trigger-happy configuration, there is no valid strategy.* □

The proofs of these propositions, found in Appendix C, all follow the same (tedious) outline: we exhaustively enumerate all possible moves that can be made in the initial configuration using the given pieces, and show that, except for the "correct" moves leading to the stated final configuration(s), each move yields an unfillable configuration, in the sense of Lemma 5.9.

Lemma 5.14 *For any $r \geq 0$, in an unprepped configuration, the only possibly valid strategy for the sequence*

$$\mathsf{I}, \mathsf{LG}, \mathsf{Sq}, r \times \langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle, \mathsf{Sq}, \mathsf{Sq}$$

is to place all of the pieces into a single bucket, yielding an unprepped configuration.

Proof. By Lemma 5.11, the only valid moves for the initial $\langle \mathsf{I}, \mathsf{LG}, \mathsf{Sq} \rangle$ place them all in some bucket, making it overflat. By induction on r and by Lemma 5.12, each successive $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ must be placed into the same bucket, yielding either a one-overflat or one-tall-plateau-one-trigger-happy configuration. Furthermore, if a one-tall-plateau-one-trigger-happy arises, then the next pieces (another $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ or a $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$) cannot be validly placed. By Lemma 5.13, the final pieces $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$ must go into the same bucket, making it—and the configuration—unprepped. □

Proposition 5.15 *Consider an unprepped configuration with exactly $4s$ total unfilled gridsquares in all buckets. Then there is a winning strategy for s successive I's only if there are exactly four unfilled gridsquares per bucket, and this strategy is to place one in each bucket to fill it up to the $(6T + 22)$nd row.*

Proof. If there are fewer than four unfilled gridsquares in any bucket, then the initial unprepped configuration must have a filled gridsquare above the $(6T+22)$nd row. Thus there is a valid strategy only if there are exactly four unfilled gridsquares in each bucket.

In this configuration, any placement of the I's other than one per bucket would fill a gridsquare beyond the $(6T+22)$nd row. □

Theorem 5.16 (Soundness) *If there is a valid strategy for $\mathcal{G}(\mathcal{P})$, then $\mathcal{P}$ is a "yes" instance of* 3-Partition.

Proof. If there is a valid strategy for $\mathcal{G}(\mathcal{P})$, then by Corollary 5.6, there is a way of placing all pieces preceding the RG into buckets to exactly fill all the empty bucket gridsquares. By Lemma 5.14, for each number a_i, we must place all of the pieces associated with a_i into a single bucket, yielding an unprepped bucket. Thus, after the pieces associated with $a_1, \ldots, a_{3s}$ have been placed, the result is an unprepped configuration with a total of $4s$ unfilled gridsquares in the buckets. By Proposition 5.15, we place exactly one I in each of these buckets to fill it up to the $(6T+22)$nd row. Thus after the numbers, each bucket must have been completely filled up to the $(6T+18)$th row, with two of the three columns filled up to the $(6T+22)$nd row.

Define A_j to be the set of a_i's so that all the pieces associated with a_i are placed into bucket j. The number of unfilled gridsquares in bucket j, by Fact 5.1, is $20T+64$. Thus the number of gridsquares filled by the pieces associated with $a_i \in A_j$ must be exactly $20T+60$, and the last four gridsquares in the bucket must have be filled by the I. The total number of gridsquares associated with the sequence corresponding to a_i is $4 \cdot [3+5a_i+2] = 20a_i+20$, by Fact 5.2. Thus the total number of gridsquares associated with numbers $a_i \in A_j$ is $\sum_{a_i \in A_j}(20a_i+20)+4$. Thus $20T+60 = \sum_{a_i \in A_j}(20a_i+20)$. Recall from Section 3 that we chose the a_i's and T so that the following facts hold:

1. if $\sum_{a_i \in A_j} a_i = T$ then $|A_j| = 3$;
2. if $\sum_{a_i \in A_j} a_i \neq T$ then $\left|T - \sum_{a_i \in A_j} a_i\right| \geq 3s$.

Then $20T+60 = \sum_{a_i \in A_j}(20a_i+20)$ implies that $\sum_{a_i \in A_j} a_i = T$ and thus that $|A_j| = 3$.

This holds for all j, so the sets $A_1, \ldots, A_s$ are a valid 3-partition. Thus $\mathcal{P}$ is indeed a "yes" instance of 3-Partition. □

Theorem 5.17 Tetris[max-cleared-rows] *is* NP*-complete.*

Proof. Immediate from Lemmas 2.2 and 5.10 and Theorems 2.1, 3.1, 3.2, 4.1, and 5.16. □

6 NP-Completeness for Other Objectives and Inapproximability

In our original definition, we considered the maximization of the number of rows cleared over the course of play. This is a fundamental component of a player's score, but in fact the score may be more closely aligned with the number of *tetrises*—that is, the number of times during play that four rows are cleared simultaneously, by the vertical placement of an I—that occur during play. (Maximizing tetrises is a typical goal in real play.)

Another type of metric—considered by Brzustowski [1] and Burgiel [2], for example—is that of *survival.* How many pieces can be placed before a loss must occur?

Define k-tetrises$(\mathcal{G}, \Sigma)$, h-height-filled$(\mathcal{G}, \Sigma)$, and p-placed-pieces$(\mathcal{G}, \Sigma)$, respectively, as the following objectives: in the game $\mathcal{G}$, does Σ, respectively, contain at least k tetrises, never fill a gridsquare above height h, and place at least p pieces before losing?

In this section, we describe reductions extending that of Section 3 to establish the hardness of optimizing each of these objectives. In Section 6.3, we give results on the hardness of approximating the number of rows cleared and the number of pieces survived. Note that all of these objectives are checkable and acyclic, and therefore the corresponding problems are in NP.

6.1 Maximizing Tetrises

We use a reduction very similar to that of Section 3, as shown in Figure 9(a). The new game is as follows:

- The top $6T + 22 + 3s + O(1)$ rows of the gameboard are exactly the same as in our previous reduction. We add four rows below these, entirely full except in the sixth column.
- The piece sequence is exactly the same as in the previous reduction, with a single I appended.

Our gameboard, shown in Figure 9(a), is that of Section 3, augmented with four new bottom rows that are full in all but the sixth column. We append a single I to our previous piece sequence.

For a "yes" instance of 3-Partition—namely, one in which we can clear the top $6T + 22$ rows using the original part of the piece sequence—$(6T + 20)/4 + 1$ tetrises are achievable. (The last occurs when the appended I is placed into the new bottom rows.)

For a "no" instance, we cannot clear the top $6T + 22$ rows using the original pieces, and since the sixth column is full in all of the original rows, we cannot clear the bottom four rows with the last I in the sequence. Thus we clear at most $6T + 22$ rows. This implies that there were at most $(6T + 20)/4 < (6T + 20)/4 + 1$ tetrises.

Therefore we can achieve $(6T + 24)/4$ tetrises just in the case that the top $6T + 22$ rows can be cleared by the first part of the sequence, which occurs exactly when the 3-Partition instance is a "yes" instance. Therefore it is NP-hard to maximize the number of tetrises achieved.

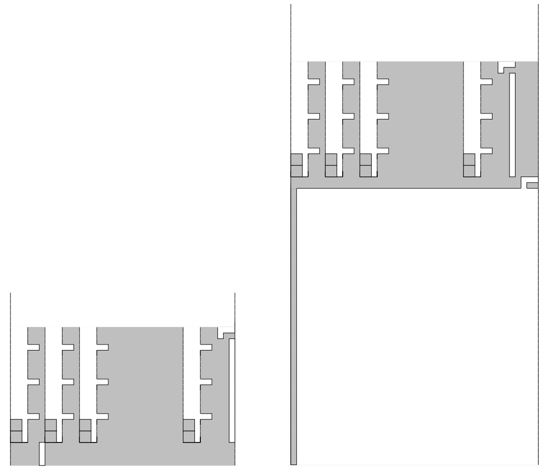
Theorem 6.1 Tetris[max-tetrises] *is* NP*-complete.* □

6.2 Maximizing Lifetime

Our original reduction yields some initial intuition on the hardness of maximizing lifetime. In the "yes" case of 3-Partition, there is a trajectory sequence that fills no gridsquares above the $(6T + 22)$nd row, while in the "no" case we must fill some gridsquare in the $(6T + 23)$rd row:

Theorem 6.2 Tetris[min-height-filled] *is* NP*-complete.* □

However, this does not immediately imply the hardness of maximizing the number of pieces that the player can place without losing, because Theorem 6.2 only applies for certain heights—and, in particular, does not apply for height m (the top row of the gameboard), because our trajectory sequence from Section 4 requires some operating space above the $(6T + 22)$nd row for rotations and translations.



(a) (b)

Figure 9: The initial gameboards showing hardness for (a) maximizing tetrises and (b) maximizing survival time. The top of these gameboards is an exact reproduction of our previous reduction.

To show the hardness of maximizing survival time, we need to do some more work. We augment our previous reduction as shown in Figure 9(b). Intuitively, we have created a large area at the bottom of the gameboard that can admit a large number of Sq's, but we place a lock so that Sq's can reach this area only if the gameboard of the original reduction is cleared. Crucially, the gameboard has odd width, so after a large number of Sq's a loss must occur.

Our new gameboard consists of the following layers, for a value r to be determined below:

- The top $6T + 22 + 3s + O(1)$ rows are exactly the same as in our previous reduction, with the addition of four completely-filled columns on the right-hand side of the gameboard.
- The two next-highest rows form a second *lock*, preventing access to the rows beneath. This lock requires a RG to be unlocked, just as in the lock columns at the top of the previous reduction. The unfilled squares in the lock are in the four new columns.
- The bottom r rows form a *reservoir*, and are empty in all columns but the first.

The gameboard has $6T + 3s + r + O(1)$ rows and $6s + 7$ columns. Let $A = O(Ts^2)$ be the total area in and above the lock rows, and let $R = r(6s + 6)$ be the total initially unfilled area in the reservoir.

Our piece sequence is augmented as follows: first we have all pieces of our original reduction, then a single RG, and finally $R/4$ successive Sq's.

For the moment, choose $r = \text{poly}(T, s)$ so that $R \geq 2A + 1$.

In the "yes" case of 3-Partition, the first part of the sequence can be used to entirely clear the $6T + 22$ rows of the original gameboard. The RG clears the second lock, and the $R/4$ successive Sq's can then be packed into the reservoir to clear all of the reservoir rows.

In the "no" case of 3-Partition, the first part of the sequence cannot entirely clear the top $6T + 22$ rows of the gameboard. Since all rows above the second lock are filled, this means that the RG cannot unlock the reservoir, and crucially the RG is the last chance to do so—no number of Sq's can ever subsequently clear the lock rows. We claim that within $A/2+1$ Sq's (which cover $2A+4$ gridsquares), a loss will occur. Since there are an odd number of columns, only rows that initially contain an odd number of filled gridsquares can be cleared by a sequence of Sq's; thus each row can be cleared at most once in the Sq sequence. In order to survive $2A+4$ gridsquares from a Sq sequence, at least one row must be cleared more than once. Therefore after $A/2+1 \leq R/4$ successive Sq's, a loss must occur.

Theorem 6.3 TETRIS[max-placed-pieces] *is* NP*-complete.* □

6.3 Hardness of Approximation

By modifying the reduction of Theorem 6.3, we can prove extreme inapproximability for either maximizing the number of rows cleared or maximizing the number of pieces placed without a loss, and a weaker inapproximability result for minimizing the maximum height of a filled gridsquare.

Theorem 6.4 *Given a sequence of p pieces, approximating* TETRIS[max-placed-pieces] *to within a factor of $p^{1-\varepsilon}$ for any constant $\varepsilon > 0$ is* NP*-hard.*

Proof. Our construction is as in Figure 9(b), but with a larger reservoir: choose r so that the r-row reservoir's unfilled area R is larger than $(2A)^{1/\varepsilon}$, where A is the total area of the gameboard excluding the reservoir rows. As before, we append to the original piece sequence one RG followed by exactly enough Sq's to completely fill the reservoir. As in Theorem 6.3, in the "yes" case of 3-PARTITION, we can place all of the pieces in the given sequence (which in total cover an area of at least R), while in the "no" case we can place pieces covering at most $2A$ area before a loss must occur. Thus it is NP-hard to distinguish the case in which we can survive all p pieces of the original sequence from the case in which we can survive at most $2A/4 < (R^{\varepsilon})/4 < p^{\varepsilon}$ pieces. □

Theorem 6.5 *Given a sequence of p pieces, approximating* TETRIS[max-cleared-rows] *to within a factor of $p^{1-\varepsilon}$ for any constant $\varepsilon > 0$ is* NP*-hard.*

Proof. Our construction is again as in Figure 9(b), with $r > a^{2/\varepsilon}$ rows in the reservoir, where there are a total rows at or above the second lock. As above, in the "yes" case of 3-PARTITION, we can completely fill and clear the gameboard, and in the "no" case we can clear at most a rows. Thus it is NP-hard to distinguish the case in which at least r rows can be cleared from the case in which at most $a < r^{\varepsilon/2}$ rows can be cleared.

Note that the number of columns c in our gameboard is fixed and independent of r, and that the number of pieces in the sequence is constrained by $r < p < (r+a)c$. We also require that r be large enough that $p < (r+a)c < r^{2/(2-\varepsilon)}$. (Note that r, and thus our game, is still polynomial in the size of the 3-PARTITION instance.) Thus in the "yes" case we clear at least $r > p^{1-\varepsilon/2}$ rows, and in the "no" case we clear at most $a < r^{\varepsilon/2} < p^{\varepsilon/2}$. Thus it is NP-hard to approximate the number of cleared rows to within a factor of $(p^{1-\varepsilon/2})/(p^{\varepsilon/2}) = p^{1-\varepsilon}$. □

Theorem 6.6 *Given a sequence of p pieces, approximating* TETRIS[min-height-filled] *to within a factor of $2-\varepsilon$ for any constant $\varepsilon > 0$ is* NP*-hard.*

Proof. Once again, our construction follows Figure 9(b). Let $F = O(Ts)$ be the total number of filled gridsquares at or above the rows of the lower lock, and let $P = O(Ts)$ be the total number of gridsquares in the piece sequence up to and including the second RG. Choose $r = (F+P)/\delta$, where $\delta = \varepsilon/(3-\varepsilon)$.

As before, in the "yes" instance of 3-PARTITION, we can place the pieces of the given sequence so that the highest filled gridsquare is in the $(6T+22)$nd row of the original gameboard, which is height $6T + 24 + r \le r + P + F \le r(1+\delta)$ in our gameboard.

In the "no" case, all of the Sq's appended to the piece sequence will have to be placed at or above the second lock, since we can never break into the reservoir. Note that, if rows are not cleared, we can never pack the appended Sq's into fewer than r rows. Thus the height of the highest filled gridsquare is at least $2r - \kappa$, where κ is the number of rows cleared during the sequence. As before, we can only clear rows that have an odd number of filled gridsquares in them before the Sq's in the sequence. Since there are only $F + P$ gridsquares in total in this part of the sequence, obviously $\kappa \le F + P = r\delta$. Thus there is a filled gridsquare at height at least $r(2-\delta)$.

Therefore it is NP-hard to approximate the minimum height of the maximum filled gridsquare to within a factor of $r(2-\delta)/r(1+\delta) = (2-\delta)/(1+\delta) = 2 - \varepsilon$. $\square$

7 Varying the Rules of Tetris

The completeness of our reduction does not depend on the full set of allowable moves in Tetris, and the soundness does not depend on all of its limitations. Thus our results continue hold in some modified settings.

7.1 Limitations on Player Agility

We have phrased the rules of Tetris so that the player can, in principle, make infinitely many translations or rotations before moving the piece down to the next-highest row. When actually playing Tetris, there is a fixed amount of time (varying with the difficulty level) in which to make manipulations at height h; one cannot slide pieces arbitrarily far to the left or right before the piece falls.

Our reduction requires only that the player be able to make two translations before the piece falls by another row (or is fixed), to slide a LG into a notch. This is why we have chosen to have $3s + O(1)$ empty rows at the top of the game board—this gives us enough room to do any desired translation and rotation before the piece reaches the top of a bucket while still only making two moves at any given height. (In the "no" case for loss-avoidance, this may cause the game to end more quickly, but the "yes" case remains feasible.)

Thus the problem remains NP-hard even when move sequences are restricted to at most two moves between drops, for any of the objectives.

7.2 Piece Set

Our reduction uses only the pieces $\{\mathsf{LG}, \mathsf{LS}, \mathsf{I}, \mathsf{Sq}, \mathsf{RG}\}$, so Tetris remains NP-complete when the pieceset is thus restricted. By taking the mirror image of our reduction, the hardness also holds for the pieceset $\{\mathsf{RG}, \mathsf{RS}, \mathsf{I}, \mathsf{Sq}, \mathsf{LG}\}$. In fact, the use of the RG in the lock was not required; we simply need some piece that does not appear elsewhere in the piece sequence. Thus Tetris remains NP-hard for any piece set consisting of $\{\mathsf{LG}, \mathsf{LS}, \mathsf{I}, \mathsf{Sq}\}$ or $\{\mathsf{RG}, \mathsf{RS}, \mathsf{I}, \mathsf{Sq}\}$, plus at least one other piece. (The reduction works exactly as before if the key for the lock is T, pointing downward; for a snake, the bottom three gridsquares of a vertically-oriented snake serves as the key, and we observe that no other piece—except T, which is not in our sequence—can be placed without filling at least *two* gridsquares above the $(6T+22)$nd row.)

7.3 Losses

We defined a loss as the fixing of a piece so that it does not fit entirely within the gameboard; i.e., the piece fills some gridsquare in the would-be $(m+1)$st row of the m-row gameboard. Instead, we might define losses as occurring only *after* rows have been cleared—that is, a piece can be fixed so that it extends into the would-be $(m+1)$st row, so long as this is not the case once all filled rows are cleared. Since the completeness sequence (of Lemma 4.1) never fills gridsquares anywhere near the top of the gameboard, our results hold for this definition as well.

In fact, for our original reduction, we do not depend on the definition of losses at all—the completeness trajectory sequence does not near the top of the gameboard, and the soundness proof does not rely on losses. Obviously the objective of Theorem 6.3 is nonsensical without a definition of losses, but all other results still hold.

7.4 Rotation Rules

In Section 2, we specified concrete rules for the rotation of pieces around a particular fixed point in each piece. In fact, our reduction applies for a wider variety of rotation models—any reasonable rotation model, as defined in Section 5. In particular, there are two other reasonable rotation rules of interest: the *continuous* model and the *Tetris* model that we have observed in practice.

In the *continuous* (or *Euclidean*) rotation model, the rotation of a piece is around its center, as before, but we furthermore require that all gridsquares that the piece *passes through* must be unoccupied.

Lemma 7.1 *The continuous rotation model is reasonable.* □

The *Tetris rotation model*, which we have observed to be the one used in a number of actual Tetris implementations, is illustrated in Figure 10. Intuitively, this model works as follows: for each piece type, choose the smallest k so that the piece fits within a k-by-k bounding box ($k = 2$ for Sq, $k = 4$ for I, and $k = 3$ otherwise). In a particular orientation, choose the smallest k_1 and k_2 so that the piece fits in k_1-by-k_2 bounding box. Place the piece so that the k_1-by-k_2 bounding box is exactly centered in the k-by-k box. This does not in general yield a position aligned on the grid, so shift the k_1-by-k_2 bounding box to the left and up, as necessary. (Incidentally, it took us some time to realize that the "real" rotation in Tetris did not follow the instantaneous model, which is intuitively the most natural one.)

Lemma 7.2 *The Tetris rotation model is reasonable.* □

The proofs of Lemmas 7.1 and 7.2, found in Appendix A, are straightforward checks of the four conditions of reasonable rotation models.

7.5 The Final Result

Theorem 7.3 *It remains* NP*-hard to optimize (or approximate) the maximum height of a filled gridsquare, the number of rows cleared, tetrises attained, or pieces placed without a loss when any of the following hold:*

1. *the player is restricted to two rotation/translation moves before each piece drops in height.*
2. *pieces are restricted to* $\{\mathsf{LG}, \mathsf{LS}, \mathsf{I}, \mathsf{Sq}\}$ *or* $\{\mathsf{RG}, \mathsf{RS}, \mathsf{I}, \mathsf{Sq}\}$ *plus at least one other piece.*
3. *losses are not triggered until* after *filled rows are cleared.*
4. *rotations follow any reasonable rotation model.*

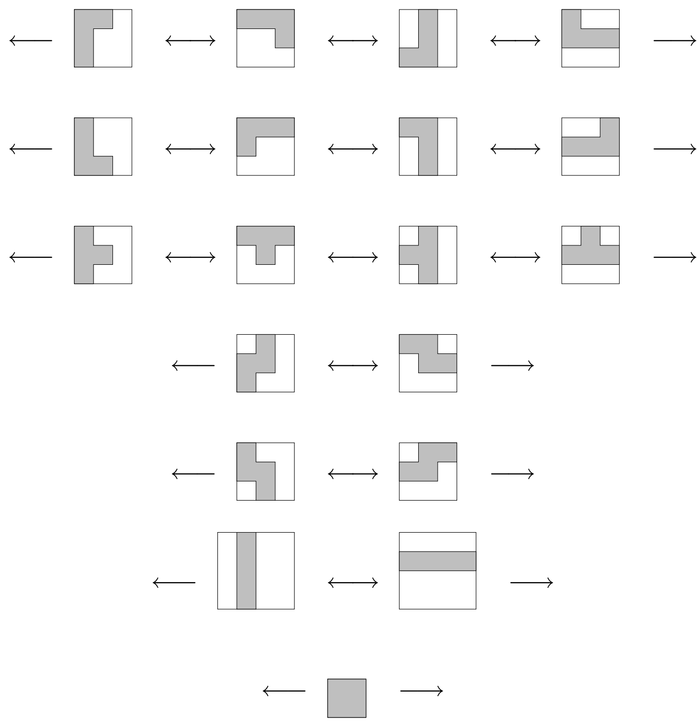


Figure 10: The Tetris model of rotation. The pictured k-by-k bounding box is in the same position in each configuration; each piece can be rotated clockwise to yield the configuration on its right (wrapping to the leftmost column) or counterclockwise to yield the configuration on its left.

When losses never occur, it remains hard to optimize (or approximate) the number of cleared rows, number of tetrises, or maximum height of a filled gridsquare. □

8 Conclusion and Future Work

An essential part of our reduction is a complicated initial gameboard from which the player must start. A major open question is whether Tetris can be played efficiently with an empty initial configuration:

- What is the complexity of Tetris when the initial gameboard is empty?

While our results hold under a variety of modifications to the rules of Tetris, some rule changes break our reduction. In particular, our completeness result relies on the translation of pieces as they fall. At more difficult levels of the game, it may be very hard to make two translations before the piece drops another row in height.

- Is translation a crucial part of the complexity of Tetris? Suppose the model for moves (following Brzustowski [1]) is the following: the piece can be translated and rotated as many times as the player pleases, and then falls into place. (That is, no translation or rotation is allowed after the piece takes its first downward step.) Is the game still hard?

Another class of open questions considers versions of the Tetris game with gameboards and piecesets of restricted size:

- What is the complexity of Tetris for a gameboard with a constant number of rows? A constant number of columns? Is Tetris fixed-parameter-tractable with respect to the number of rows or number of columns? (We have polynomial-time algorithms for the special cases in which the total number of gridsquares is logarithmic in the number of pieces in the sequence, or for the case of a gameboard with two columns.)

- We have reduced the pieceset down to five of the seven pieces. For what piecesets is Tetris polynomial-time solvable? (For example, with the pieceset $\{\mathsf{I}\}$ the problem seems polynomially solvable, though non-trivial because of the initial partially filled gameboard.)

Finally, in this paper we have concentrated our efforts on the offline, adversarial version of Tetris. In a real Tetris game, the initial gameboard and piece sequence are generated probabilistically, and the pieces are presented in an online fashion:

- What can we say about the difficulty of playing online Tetris if pieces are generated independently at random according to the uniform distribution, and the initial gameboard is randomly generated?

Acknowledgments. The first author thanks Amos Fiat and Ming-wei Wang for helpful initial discussions. Thanks also to Josh Tauber for pointing out the puzzles in *Games Magazine* [7].

References

[1] John Brzustowski. Can you win at Tetris? Master's thesis, University of British Columbia, 1992.

[2] Heidi Burgiel. How to lose at Tetris. *Mathematical Gazette*, July 1997.

[3] Erik D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *Proc. 26th Symposium on Mathematical Foundations in Computer Science*, pages 18–32, August 2001. Full version available at `http://arxiv.org/abs/cs.CC/0106019`.

[4] Michael R. Garey and David S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM J. Comput.*, 4:397–411, 1975.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.

[6] Richard Kaye. Minesweeper is NP-Complete. *Mathematical Intelligencer*, 22(2):9–15, 2000.

[7] Scott Kim. Tetris unplugged. *Games Magazine*, pages 66–67, July 2002.

[8] Michael M. Kostreva and Rebecca Hartman. Multiple objective solution for Tetris. Technical Report 670, Department of Mathematical Sciences, Clemson University, May 1999.

[9] David Sheff. *Game Over: Nintendo's Battle to Dominate an Industry.* Hodder and Stoughton, London, 1993.

[10] Tetris, Inc. `http://www.tetris.com`.

A Rotation Models

Recall that a rotation model is *reasonable* if it satisfies the following four conditions:

1. A piece cannot "jump" from one bucket to another, or into a disconnected region.
2. An LG cannot enter a notch if the gridsquare in the second column of the notch row is filled.
3. For any balcony at height h with the ith column empty in the balcony, the only gridsquares at height $h' \leq h$ that can be filled by any piece other than an I are those in the ith column in rows h and $h-1$.
4. An I cannot enter a notch if the gridsquares in the second and third columns of the row immediately beneath the notch row are filled.

As a preliminary step, we will give some sufficient conditions on rotation models for two of these conditions.

Lemma A.1 *Suppose, for every rotation of every piece, there is some gridsquare that is filled both before and after the rotation. Then clause (1) of the definition of reasonable is satisfied.*

Proof. Partition the unfilled gridsquares of the bucket into connected *regions* (where two gridsquares are connected if and only if they share an edge), counting the four gridsquares containing the current piece as unfilled. Since some gridsquare remains filled by the piece before and after a legal rotation, the four gridsquares filled before and after the rotation must be in the same region. Furthermore, neither a horizontal translation nor drop can shift the piece from one region to another.

By definition, a hole is a different region from that at the top of the bucket, where the piece enters. A piece can never move from one region to another, so therefore it can never fill a hole. Similarly, a piece can never jump from one bucket to another, since doing so would force it to pass through the wall dividing the buckets. □

Lemma A.2 *Suppose, for every rotation of every piece except* I*, there is a row h that has at least two gridsquares filled by the piece before the rotation, and, after the rotation, one of the following holds:*

(a) some row $h' \geq h-1$ still has at least two gridsquares filled by the piece.

(b) row $h-2$ has at least two gridsquares filled by the piece, and at least one gridsquare in row $h-1$ that was unfilled before the rotation is now filled.

Then clause (3) of the definition of reasonable is satisfied.

Proof. Suppose, as per clause (3), that there is a balcony at height b, with the ith column empty in row b.

For every piece other than I, in any orientation, there is a row that has at least two gridsquares filled by that piece. Call such a row *heavy*. In any configuration in which a heavy row is at height $b+1$ or above, it is easy to confirm that clause (3) is satisfied.

Suppose that there is a way to move the piece so that the following condition holds:

($*$) all of its heavy rows appear in row $b-1$ or lower.

Consider the first time that (∗) occurs, and consider the previous move σ, i.e., the one that made (∗) true.

It is obvious that σ cannot be a horizontal translation or a fix, since these moves do not affect the vertical placement of the heavy rows. Furthermore, the move σ cannot have been a drop, since the filled gridsquares in the topmost heavy row would somehow have passed through the balcony. Thus σ must have been a rotation.

If σ is a rotation satisfying condition (a), then there is a heavy row that cannot have passed below the balcony—it is either still above the balcony, or at best is in the same row as the balcony, implying that the rotation was not legal (since two of the three gridsquares in row b were already occupied).

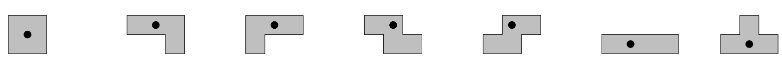
Otherwise, the rotation σ must satisfy the condition (b). The only way to have made (∗) true would be for a heavy row to have been in row $b+1$ before the rotation and $b-1$ after. But, by condition (b), to do this the piece must occupy two distinct gridsquares in row b: one before and a different one after the rotation. Since row b is a balcony, and two of its three gridsquares are already occupied, this is impossible.

Thus there is no such move σ, and clause (3) is satisfied. □

A.1 Instantaneous Rotation

Lemma 5.10 *The instantaneous rotation model is reasonable.*

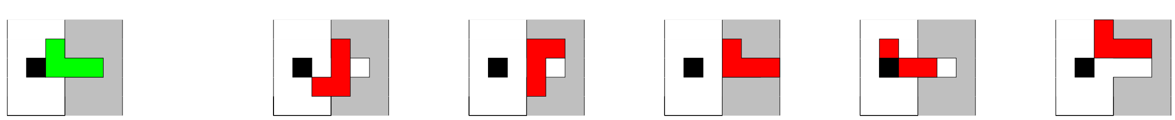
Proof. Recall that in the instantaneous rotation model, rotation occurs around the *center* of each piece:



1. **No piece can fill a hole.** Immediate from Lemma A.1—by definition, the gridsquare containing the center of the piece is filled before and after the rotation.

2. **No LG can fully fill a spurned notch.** Suppose that the LG managed to enter the notch, and consider the first time that it did so. It is in the first of the following configurations when this occurs.

   In the move before its first entry into the notch, it must be in one of the remaining configurations (respectively, arising from a clockwise rotation, counterclockwise rotation, left slide, right slide, and drop). Note that it is impossible for the LG to have entered the notch via a fix, which does not change the piece's position or orientation.

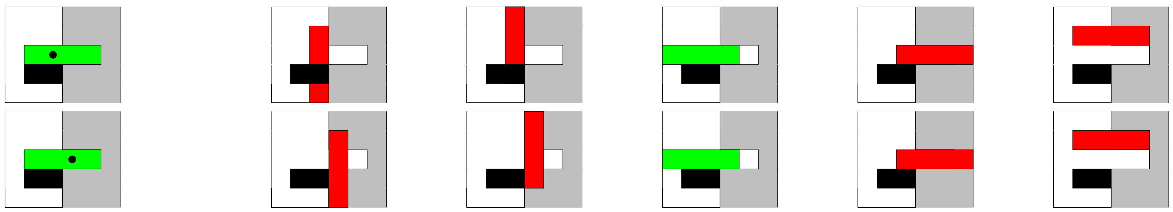
   

   In each of these configurations, the LG intersects a filled gridsquare—in the fourth, in the second column of the notch row; for the others, one or more of the initially-filled gridsquares in the notch columns of the bucket.

3. **Only an I can fully pass a balcony.** Immediate from Lemma A.2—one can easily verify that, in fact, the first stated condition applies for each rotation.

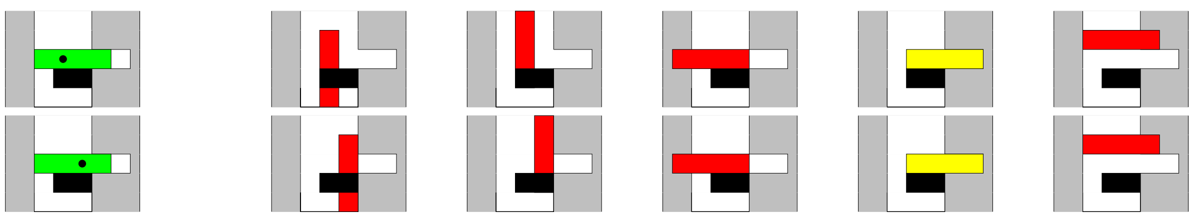
4. **An I cannot enter a 011-floored notch or 111-floored notch.** Suppose that the I managed to enter the notch, and consider the first time that it did so. It is in the first configuration of one of the following rows when this occurs. (Note that the rotation center could be in either the second or third gridsquare of the I, reading from left to right.)

   In the move before the I's first entry into the notch, it must be in one of the remaining configurations—respectively, arising from a clockwise rotation, counterclockwise rotation, left slide, right slide, and drop. Note that it is impossible for the I to have entered the notch via a fix, which does not change the piece's position or orientation.



   In each of these configurations but the third in each row, the previous position overlaps an occupied gridsquare—one of the assumed gridsquares in the first two configurations of the first row, and an initially-filled gridsquare in the notch columns in the remainder.

   Thus the only way for the I to fully enter the notch is for it to have previously been partially in the notch. Now we consider the previous position of the I before it was partially in the notch. The following configurations are arranged analogously to the above.



   But, in all but the fourth position of each row, the I overlaps a filled square, as before; in the fourth, the previous position is fully in the notch. But, by our assumption that we are considering the *first* entry into the notch, this cannot have been the previous configuration.

   Therefore the I cannot fully enter the 011-floored notch or 111-floored notch.

Thus the instantaneous model is reasonable. □

A.2 Continuous Rotation

Lemma 7.1 *The continuous rotation model is reasonable.*

Proof. Any move which is allowed in the continuous rotation model is also allowed in the instanteous rotation model, so this follows immediately from Lemma 5.10. □

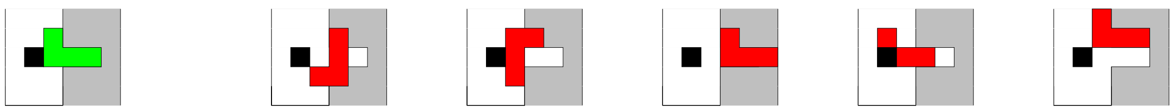
A.3 Tetris Rotations

Lemma 7.2 *The Tetris rotation model is reasonable.*

Proof. We show that the four requirements are met:

1. **No piece can fill a hole.** Immediate from Lemma A.1 and inspection of the rotations.

2. **An LG cannot enter a spurned notch.** Suppose that the LG somehow managed to enter the notch. Then it must be in the first of the following configurations after it enters the notch, and must have previously been in either the remaining configurations immediately before (respectively, arising from a clockwise rotation, counterclockwise rotation, left slide, right slide, and drop). Note that it is impossible for the LG to have entered the notch via a fix, which does not change the piece's position or orientation.

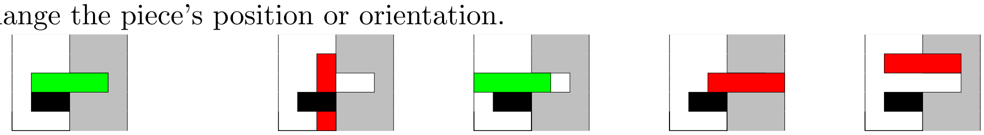
   

   In each of these configurations, the LG intersects a filled gridsquare—in the fourth, in the second column of the notch row; for the others, one or more of the initially-filled gridsquares in the notch columns of the bucket.

3. **Only an I can fully pass a balcony.** Immediate from Lemma A.2 and inspection of the rotation rules.

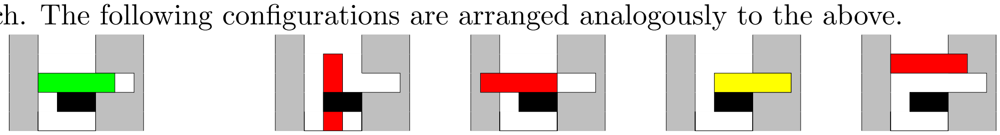
4. **An I cannot enter a 011-floored notch or 111-floored notch.** This is analagous to this condition for instantaneous rotation, but simpler.

   Suppose that the I managed to enter the notch, and consider the first time that it did so. It is in the first configuration of the following when this occurs. In the move before the I's first entry into the notch, it must be in one of the remaining configurations—respectively, arising from a rotation (clockwise and counterclockwise have the same result), right slide, left slide, and drop. Note that it is impossible for the I to have entered the notch via a fix, which does not change the piece's position or orientation.

   In each of these configurations but that arising from a right slide, the previous position overlaps an occupied gridsquare.

   Thus the only way for the I to fully enter the notch is for it to have previously been partially in the notch. Now we consider the previous position of the I before it was partially in the notch. The following configurations are arranged analogously to the above.

   In all but the position arising from the left slide, the I overlaps a filled square, as before. By our assumption that we are considering the *first* entry into the notch, this cannot have been the previous configuration.

   Therefore the I cannot fully enter the 011-floored notch or 111-floored notch.

Thus the Tetris model is reasonable. □

B Deriving Unfillability from Reasonable Rotation Models

Recall that a bucket is *unfillable* if it cannot be filled completely using arbitrarily many pieces from the set $\{\mathsf{LG}, \mathsf{LS}, \mathsf{Sq}, \mathsf{I}\}$. Note that, for the purposes of proving unfillability, we do not need to worry about the piece RG.

In this appendix, we show that a variety of configurations are unfillable in any reasonable rotation model. Note that by clause (1) of reasonable, we can limit our attention to a single bucket when proving unfillability.

Lemma B.1 *In any reasonable rotation model, a bucket with (a) a hole, (b) a spurned notch, (c) a balconied 011-floored notch, or (d) a balconied 111-floored notch is unfillable.*

Proof. Claim (a) is immediate from clause (1) of the definition. Since an I in the spurned notch would have to occupy the (filled) gridsquare in the second column of that row, an LG is the only piece that could fill the notch; thus claim (b) follows from clause (2) of the definition of reasonable. For claims (c) and (d), the unfilled notch must be filled with an I by clause (3), and by clause (4) no I can enter the notch. □

Lemma B.2 *In any reasonable rotation model, a bucket with a balconied 110-floored notch is unfillable.*

Proof. Say the balconied 110-floored notch is in row n. By clause (3), only an I can fill the notch in row n. If the notch is filled by an I before the empty gridsquare in the third column of row $n-1$ is filled, then the result is a hole. If the empty gridsquare in the third column of row $n-1$ is filled before the notch in row n is filled, then the result is a balconied 111-floored notch, or a hole if filling the third gridsquare in row $n-1$ simultaneously fills the third gridsquare of row n. Either way, the result is unfillable. □

Lemma B.3 *Consider a bucket with a balcony in row h, and, somewhere below row h, an α-veranda. In any reasonable rotation model, the bucket is unfillable if either*

1. *the α-veranda spans no notch rows, and $\alpha \not\equiv 0 \pmod 4$, or*
2. *the α-veranda spans one notch row, and $\alpha \notin \{0, 1\} \pmod 4$.*

Proof. By clause (3) of the definition of reasonable, the α-veranda can only be filled by I's. (The α-veranda cannot be located in the exception of clause (3)—in the ith column in the h and $(h-1)$st rows, where the balcony leaves the ith column of row h open—because otherwise the α-veranda would be above row h.)

1. An I cannot fit horizontally into any non-notch row, so the only way to fill the α-veranda is with vertically placed I's. But since $\alpha \not\equiv 0 \pmod 4$, the α-veranda cannot be completely filled by vertical I's.

2. Just as in case (1), since $\alpha \not\equiv 0 \pmod 4$ we cannot fill the α-veranda using vertical I's alone. The only other possibility is to place an I horizontally into the notch row, filling some gridsquare in the middle of the α-veranda. This results in a β-veranda and γ-veranda, where the notch was in the $(\beta+1)$th-highest row of the α-veranda and $\beta + 1 + \gamma = \alpha$.

   By the assumption that $\alpha - 1 \not\equiv 0 \pmod 4$, we know that $\beta + \gamma \not\equiv 0 \pmod 4$, which implies that at least one of β-veranda and γ-veranda falls into case (1).

Thus such a bucket is unfillable. $\square$

Lemma B.4 *In any reasonable rotation model, a bucket with a balconied* ($\alpha \not\equiv_4 0$)*-veranda in non-notch rows or balconied* 2*-veranda is unfillable.*

Proof. Immediate from Lemma B.3, since at most one of the rows spanned by a balconied 2-veranda is a notch row. $\square$

Lemma B.5 *In any reasonable rotation model, a bucket with a balconied unfilled sub-notch rectangle is unfillable.*

Proof. Suppose the notch in row n has the rectangle beneath it unfilled. By clause (3) of the definition of reasonable, only I's can be used to fill the notch in row n and the unfilled gridsquares in the rectangle below the notch.

Consider the unfilled gridsquare(s) in the ith column in rows $n-1, \ldots, n-j$, $i \in \{2, 3\}$ and $j \in \{1, 2, 3\}$, and the filled gridsquare in row $n-j-1$. These unfilled gridsquares must be filled by a vertically-placed I, since they are in non-notch rows.

If $i = 3$ and the gridsquare in column 3 and row $n-1$ is filled before the notch in row n is filled, then the result is a hole: since $j \leq 3$, the vertical I extends into row n. If $i = 2$ and the gridsquare in column 2 and row $n-1$ is filled before the notch in row n is filled, then the result is a spurned notch: again, the vertical I extends into row n.

If, on the other hand, the notch is filled before the unfilled gridsquare in row $n-1$, then the result is a j-veranda: in column i, rows n and $n-j-1$ are filled and rows $n-1, \ldots, n-j$ are unfilled. Furthermore, for $j \leq 3$, none of the unfilled rows are notch rows (since n is a notch row). This is unfillable by Lemma B.3.

Regardless of which we try to fill first, the resulting configuration is unfillable. $\square$

Lemma B.6 *In any reasonable rotation model, a bucket with a balconied flat-bottomed 2/3-ceiling discrepancy is unfillable.*

Proof. Suppose that in row f the gridsquares in both the second and third columns are filled, and in rows $f+1, \ldots, f+\alpha$ both are empty, and that the second gridsquare in row $f+\alpha+1$ is filled. Further suppose that the second column is full and the third column empty in rows $f+\alpha+1, \ldots, f+\alpha+\beta$, and is filled in row $f+\alpha+\beta+1$, where $\beta \not\equiv 0 \pmod{4}$. (The case when columns two and three are swapped is analogous.) By clause (3) of the definition of reasonable, only I's can be used to fill the unfilled gridsquares beneath row $f+\alpha+\beta+1$.

If $f+\alpha+1$ is a notch row, then it is a spurned notch—or a hole for the case in which the unfilled segment of column three is the shorter one—which is unfillable by Lemma B.1. If all of $f+1, \ldots, f+\alpha$ are non-notch rows, then we have an α-veranda and an $(\alpha+1)$-veranda spanning no notch rows. By Lemma B.3, this is unfillable. Otherwise, there is at least one unfilled notch row in rows $f+1, \ldots, f+\alpha$; consider the highest such notch row r.

If an I is placed horizontally to fill the notch in row r before any gridsquares in the second or third columns at or above row r are filled, then the resulting configuration has a balconied flat-bottomed 2/3-ceiling discrepancy spanning only non-notch rows. By the previous case, this is unfillable.

Thus some I must be placed vertically to fill gridsquares in the second or third columns at or above row r before the notch is filled. Such a placement of an I must fill either the second or third gridsquare of row r as well (a vertical I fills four unfilled gridsquares above a filled gridsquare in the same column). This creates a spurned notch or a hole, respectively.

In any case, then, the configuration is unfillable. □

Lemma B.7 *In any reasonable rotation model, a bucket with a balconied notch-spanning 2/3-ceiling discrepancy is unfillable.*

Proof. By clause (3) of the definition of reasonable, only I's can be used to fill the spanned notch. Once this notch is filled, the result is a balconied flat-bottomed 2/3-ceiling discrepancy. □

Lemma 5.9 *In any reasonable rotation model, any bucket containing any of the following is unfillable:*

1. *a hole.*
2. *a spurned notch.*
3. *a balconied 110-floored notch, 011-floored notch, or 111-floored notch.*
4. *a balconied unfilled sub-notch rectangle.*
5. *a balconied* $(\alpha \not\equiv_4 0)$*-veranda in non-notch rows.*
6. *a balconied* 2*-veranda.*
7. *a balconied flat-bottomed 2/3-ceiling discrepancy.*
8. *a balconied notch-spanning 2/3-ceiling discrepancy.*

Proof. Immediate from Lemmas B.1, B.2, B.4, B.5, B.6, and B.7. □

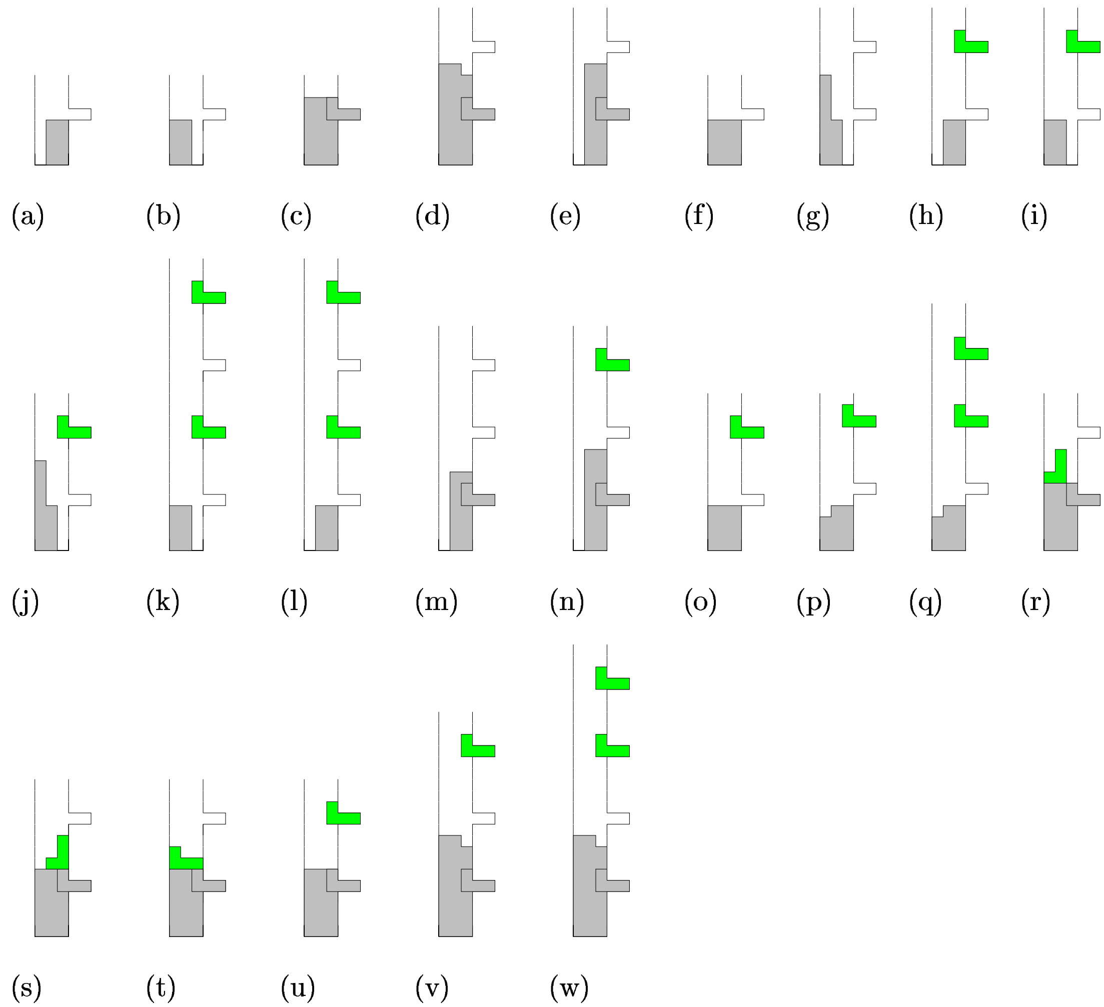


Figure 11: Configurations that can arise during play: (a,b) unprepped, (c) overflat, (d) trigger-happy (e) tall-plateau, (f) underflat, (g) (I-UP), (h,i) (LG-UP-i), (j) (I-UP-LG-i), (k,l) (LG-UP-$\{i,j\}$), (m) short-plateau, (n) (LG-TP-i), (o) (LG-UF-i), (p) (LG-UA-i), (q) (LG-UA-$\{i,j\}$), (r) (LG-OF-1), (s) (LG-OF-2), (t) (LG-OF-3), (u) (LG-OF-4-i), (v) (LG-TH-i), (w) (LG-TH-$\{i,j\}$). In each annotated configuration, the ith-highest (and jth-highest) notches have LGs filling them.

C Proof of Soundness Propositions

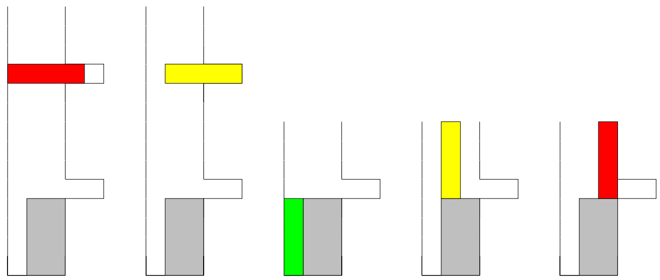
In this section, we show that, under any reasonable rotation model, there is no possible way to win the game $\mathcal{G}(\mathcal{P})$ without playing "right." In Figure 11, we show the configurations that will be relevant in our proofs.

C.1 Pieces and Configurations

C.1.1 Unprepped Buckets

Proposition C.1 *If* I *is dropped validly into an unprepped bucket, then it produces either an underflat bucket or an (*I*-UP) bucket.*

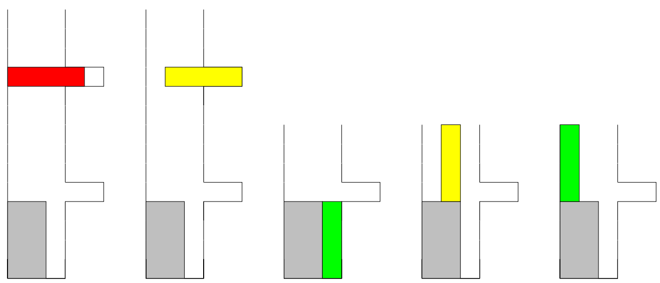
Proof. For an unprepped bucket in which the first column is the unfilled one, the possible placements of the I are as follows:



(The first and second configurations have the I horizontally in the ith-highest notch. Both of these configurations are blocked for $i = 1$, since this configuration has a 011-floored notch and thus no I can enter the bottom notch.)

The first has a hole for any i. In the second, for any $i \neq 1$, the lowest notch is a balconied 011-floored notch; for $i = 1$, this is blocked. The third configuration is underflat, the fourth configuration has a spurned notch, and the last has a hole.

For the other kind of unprepped bucket, with the third column unfilled, the possible placements of the I are as follows:

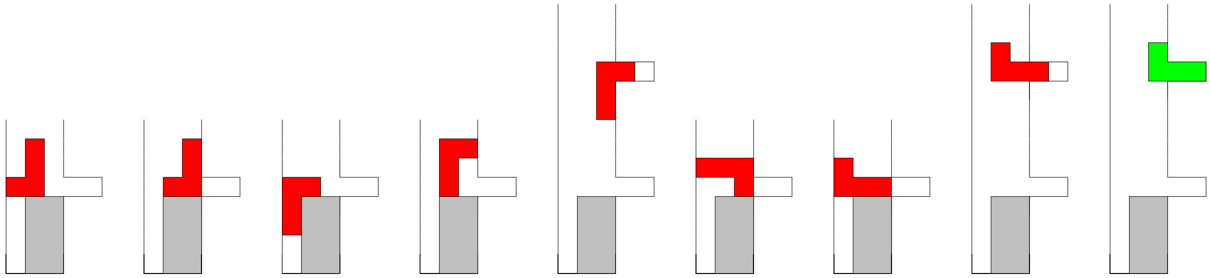


(The first and second configurations have the I horizontally in the ith-highest notch.)

The first has a hole in the notch for any i. The second configuration has a hole for $i = 1$ and is a balconied 110-floored notch for $i > 1$. The third is underflat, and the fourth has a spurned notch. The fifth is (I-UP). ☐

Proposition C.2 *If* LG *is dropped validly into an unprepped bucket, it produces (*LG*-UP-i) for some i.*

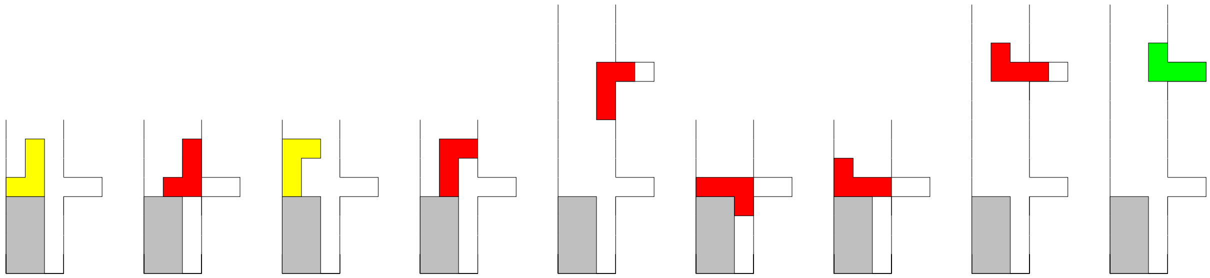
Proof. If the unprepped bucket has the first column empty and the bottom four rows of the second and third columns full, then the possible configurations are the following:



(In the fifth configuration, the ith-highest notch is partially filled, for any $i > 1$; the eighth and ninth apply for any $i \geq 1$.)

Each but the last of these configurations creates a hole (the fifth for any $i > 1$ and the eighth for any $i \geq 1$); the last is (LG-UP-i).

For the other kind of unprepped bucket, with an empty third column, the possible configurations are the following:

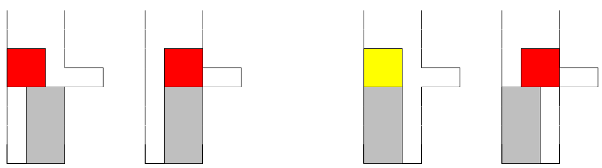


(The fifth, eighth, and ninth have the LG in the ith-highest notch; these apply for any i.)

The first configuration has a spurned notch, the third has a balconied 110-floored notch, and the last is (LG-UP-i). The remainder of these configurations all have holes (the fifth and eighth for all $i \geq 1$). □

Proposition C.3 *There is no valid move for* Sq *in an unprepped bucket.*

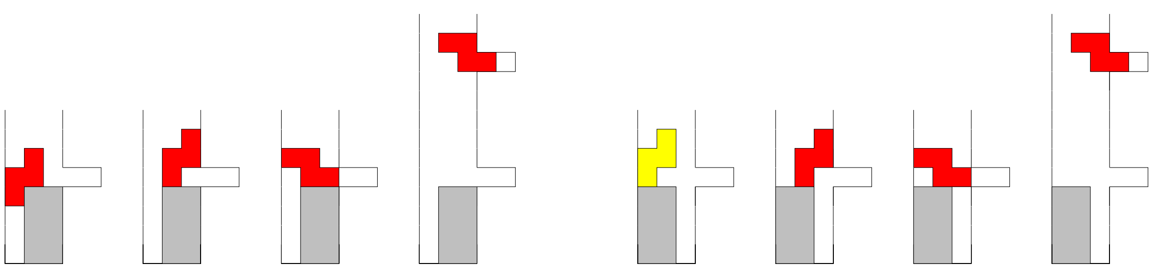
Proof. The possible placements for a Sq are as follows:



The first, second, and fourth of these create holes; the third has a spurned notch. □

Proposition C.4 *There is no valid move for* LS *in an unprepped bucket.*

Proof. The possible configurations are the following:



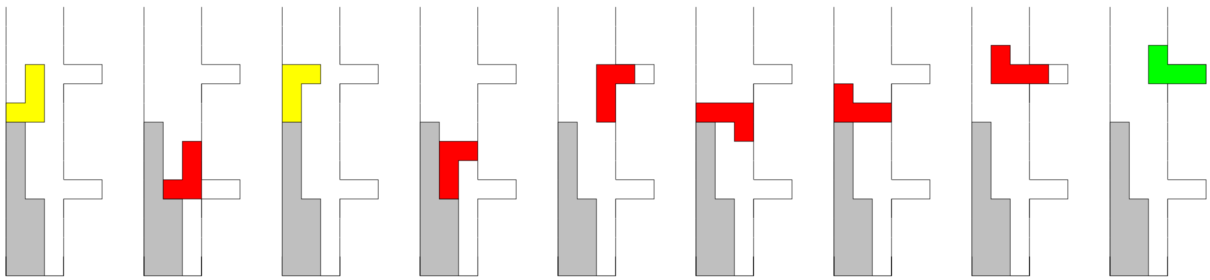
(In the fourth and eighth of these configuration, the LS is in the ith-highest notch, for any $i \geq 1$.)

All of these configurations except the fifth have holes; the fifth has a balconied 110-floored notch. □

C.1.2 (I-UP) Buckets

Proposition C.5 *If* LG *is dropped validly into an (*I*-UP) bucket, it produces (*I*-UP-*LG*-i) for some $i \geq 2$.*

Proof. The possible configurations are the following:

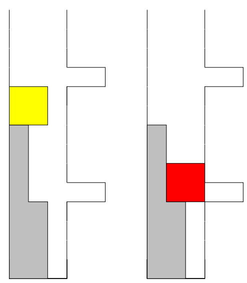


(In the fifth, eighth, and ninth configurations, the ith-highest notch is (partially) filled, for any $i \geq 1$.)

The first has a balconied 110-floored notch, as does the third. All other configurations except the last have holes (including the fifth and eighth for any i). The last is (I-UP-LG-i), though there is a hole if $i = 1$. □

Proposition C.6 *There is no valid placement of* Sq *in an (*I*-UP) bucket.*

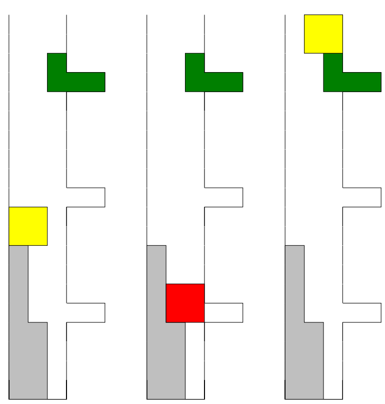
Proof. The possible configurations are the following:



The first has a balconied 110-floored notch, and the second has a hole. □

Proposition C.7 *There is no valid placement of* Sq *in an (*I*-UP-*LG*-i) bucket.*

Proof. The possible configurations are the following:



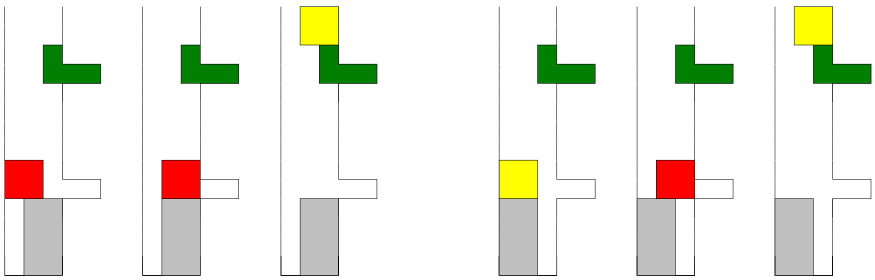
(In each configuration, the LG is initially in the ith-highest notch for any $i > 1$; if $i = 1$, the configuration already has a hole.)

The first and third configurations have a balconied 110-floored notch; the second has a hole. □

C.1.3 LG-prepped Buckets

Proposition C.8 *There is no valid move for* Sq *in (*LG*-UP-i).*

Proof. The possible configurations are as follows:

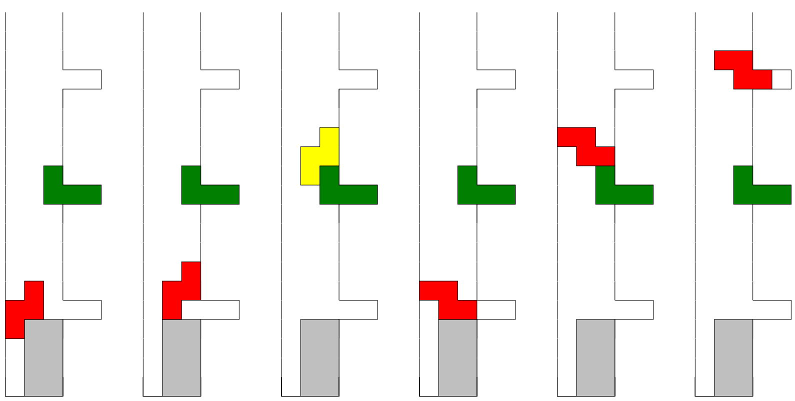


(In each configuration, the LG is initially in the ith-highest notch for any $i \geq 1$. The second and fifth configurations are blocked if $i = 1$.)

The first, second (for $i > 1$), and fifth (for $i > 1$) configurations have holes. The third has a balconied 011-floored notch if $i > 1$, or a balconied 2-veranda if $i = 1$. The fourth has a spurned notch if $i > 1$, and a hole if $i = 1$. The sixth has a balconied 110-floored notch if $i > 1$, or a hole if $i = 1$. □

Proposition C.9 *There is no valid move for* LS *in (*LG*-UP-i).*

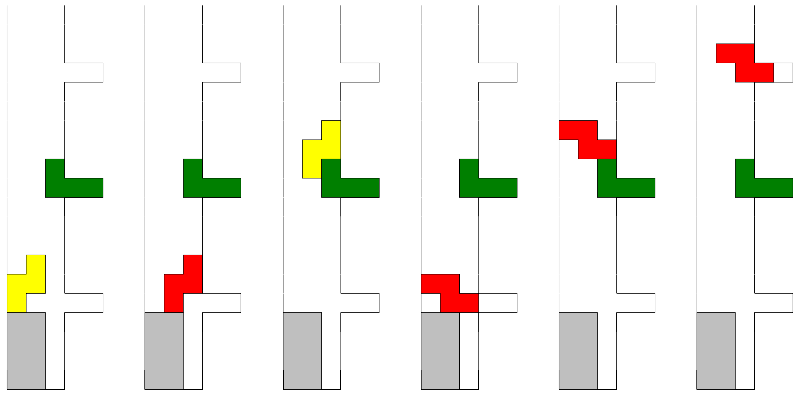
Proof. With (LG-UP-i) so that the first column is unfilled in the bottom four rows, the possible configurations are the following:



(The LG is in the ith-highest column initially. In the last of these configurations, the LS is in the jth-highest notch, for any $j \geq 1$ and $j \neq i$. The second and fourth configurations are possible only if $i \neq 1$.)

Of these, all configurations but the third create holes (the last for any j), and the third has a balconied 011-floored notch, or a balconied 1-veranda in non-notch rows for $i = 1$.

With (LG-UP-i) so that the first four rows of the third column are unfilled, the possible configurations are the following:

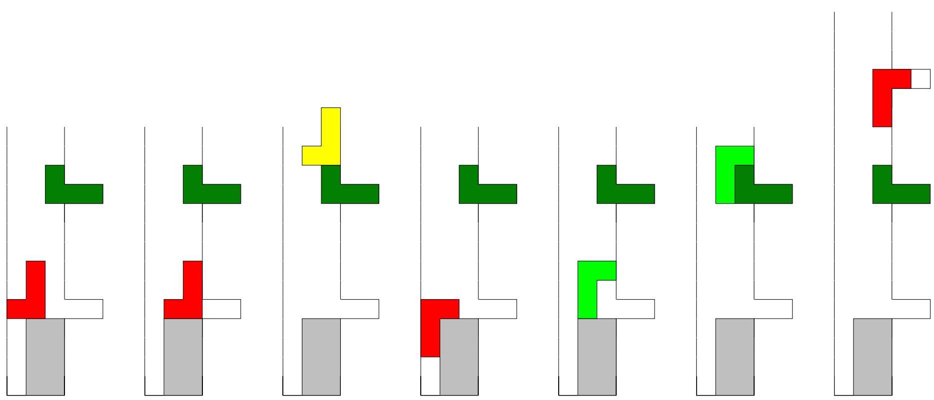


(Again, the LG is in the ith-highest notch initially and the LS in the last configuration is in the jth-highest notch, for any $j \geq 1$ and $j \neq i$. The second and fourth configurations are possible only if $i \neq 1$.)

The first and third both have a balconied 110-floored notch, or a hole if $i = 1$. The second (for $i > 1$), fourth, fifth, and sixth (for any j) have a hole. □

Proposition C.10 *If* LG *is validly placed in (*LG*-UP-*i*), then the result is a short-plateau or a (*LG*-UP-*$\{i, j\}$*) for some* j*.*

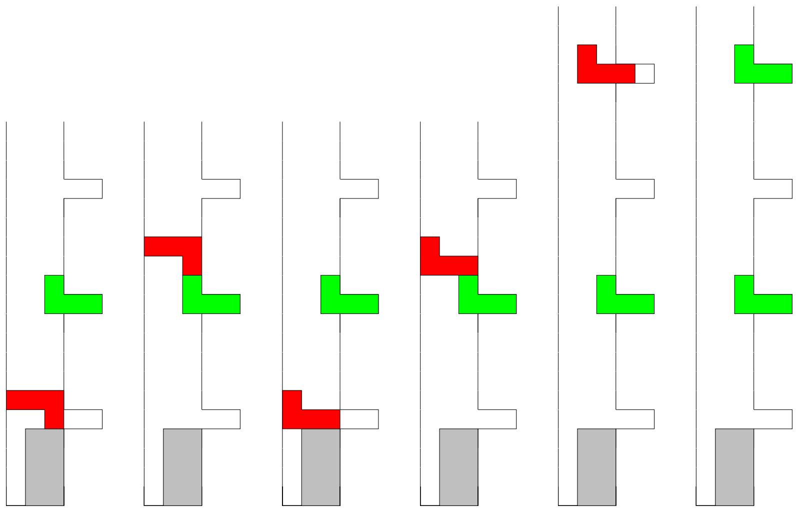
Proof. When the given (LG-UP-i) bucket has the empty rows in the first column, the possible configurations are as follows. First, the configurations in which the LG is placed vertically:



(The notch initially filled with the LG is the ith-highest; in the last configuration, the jth-highest notch is also filled for any $j \neq i$ and $j \geq 1$. If $i = 1$, the second configuration is impossible, and the fifth and sixth configurations are identical.)

The first, second, fourth, fifth (for $i > 1$), and seventh (for any $j \neq i$) configurations have holes. The third has a balconied 2-veranda for $i = 1$ and a balconied 011-floored notch for $i > 1$. The sixth also has a balconied 011-floored notch for $i > 1$. For $i = 1$, the fifth and sixth configurations are identical, and form a short-plateau.

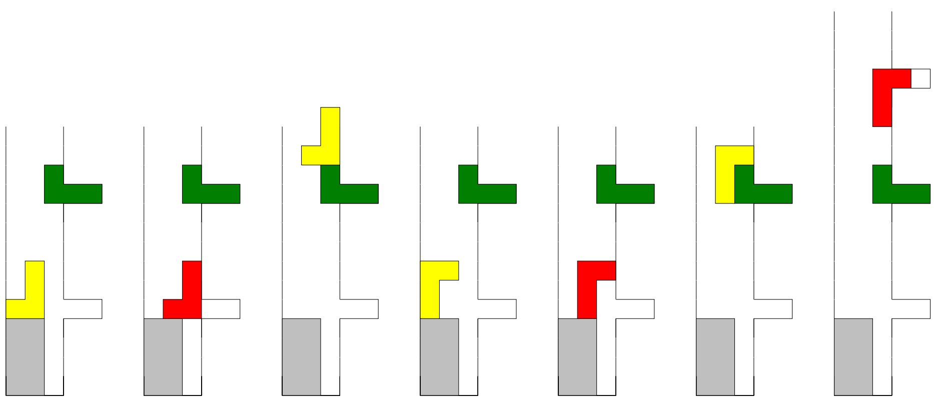
And, now, those in which the LG is placed horizontally:



(The LG is initially in the ith-highest notch; in the last two configurations, the jth-highest notch is also (partially) filled by a new LG, for any $j \neq i$ and $j \geq 1$. If $i = 1$, the first and third configurations are impossible.)

All but the last of these has a hole (and the fifth has a hole regardless of the value of j). The last configuration is (LG-UP-$\{i, j\}$).

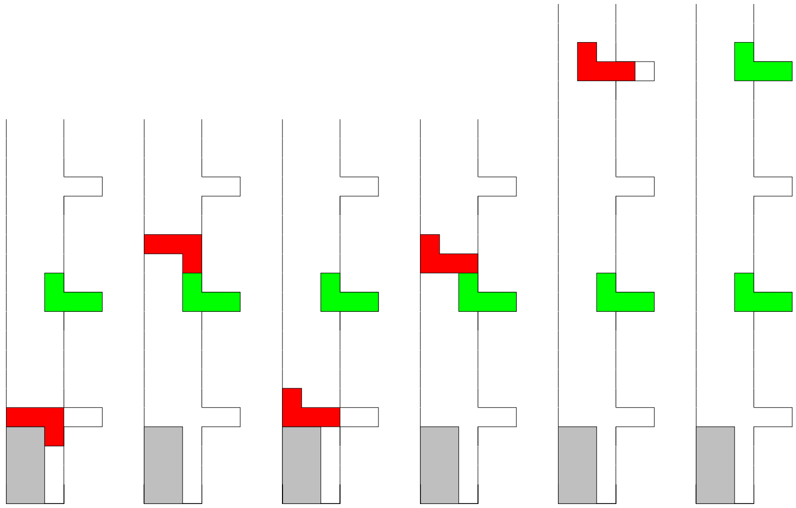
When the given (LG-UP-i) bucket has the empty rows in the third column, the possible configurations are as follows. When the LG is placed vertically:



(The notch initially filled with the LG is the ith-highest for any $i \geq 1$; in the last configuration, the jth-highest notch is also filled for any $j \neq i$ and $j \geq 1$. If $i = 1$, then there is a hole initially, and all configurations are invalid.)

The first has a spurned notch. The second, fifth, and seventh (for any j) have holes. The third, fourth, and sixth all have a balconied 110-floored notch.

And, now, those in which the LG is placed horizontally:

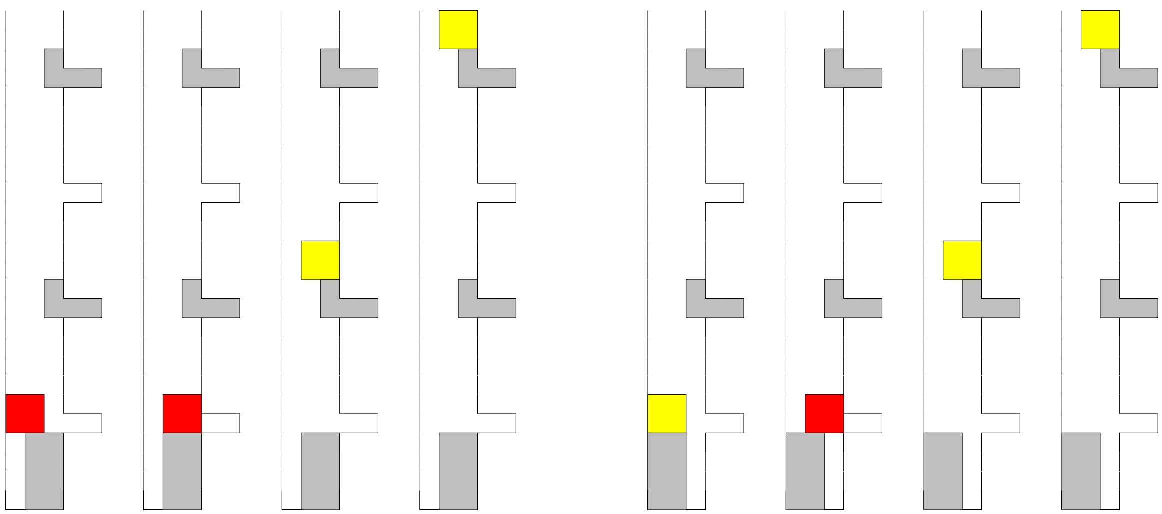


(The LG is initially in the ith-highest notch for any $i \geq 1$; in the last two configurations, the jth-highest notch is (partially) filled by a new LG, for any $j \neq i$. If $i = 1$, there is a hole initially, and all configurations are invalid.)

All but the last of these has a hole (and the next-to-last has a hole regardless of the value of j). The last configuration is (LG-UP-$\{i, j\}$). □

Proposition C.11 *There is no valid move for* Sq *in (*LG*-UP-*$\{i, j\}$*).*

Proof. The possible configurations are:



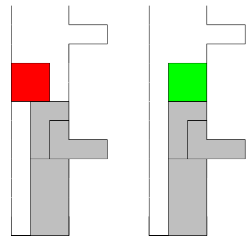
(The initially filled notches are the ith- and jth-highest, for $i, j \geq 1$ and $i \neq j$. The second configuration is blocked for $\min(i, j) = 1$. The fifth, sixth, seventh, and eighth configurations have holes and are thus invalid if $\min(i, j) = 1$.)

In the first, second, and sixth configurations, there are holes. The third has a balconied 2-veranda for $\min(i, j) = 1$ and a balconied 011-floored notch for $\min(i, j) > 1$. The fourth has a balconied 5-veranda in non-notch rows for $\max(i, j) = 2$, a balconied notch-spanning 2/3-ceiling discrepancy for $\min(i, j) = 1$ and $\max(i, j) > 2$, and a balconied 011-floored notch for $\min(i, j) > 1$. In the fifth configuration, there is a spurned notch. The seventh and eighth each have a balconied 110-floored notch for $\min(i, j) > 1$. □

C.1.4 Short-Plateau Buckets

Proposition C.12 *If* Sq *is validly placed in a short-plateau bucket, then the result is a tall-plateau.*

Proof. The only possible configurations are

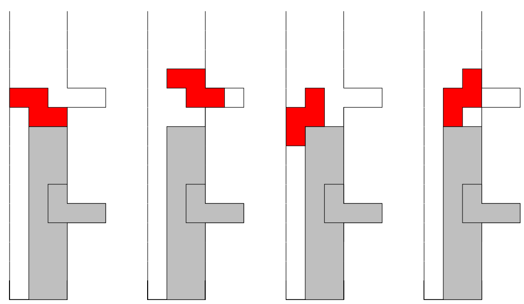


The first has a hole; the second is a tall-plateau, as desired. □

C.1.5 Tall-Plateau Buckets

Proposition C.13 *There is no valid move for* LS *in a tall-plateau bucket.*

Proof. The possible configurations are the following:

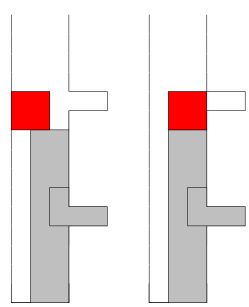


(The second configuration has the LS in the ith-highest notch, for any $i \geq 1$.)

All four have holes. □

Proposition C.14 *There is no valid move for* Sq *in a tall-plateau bucket.*

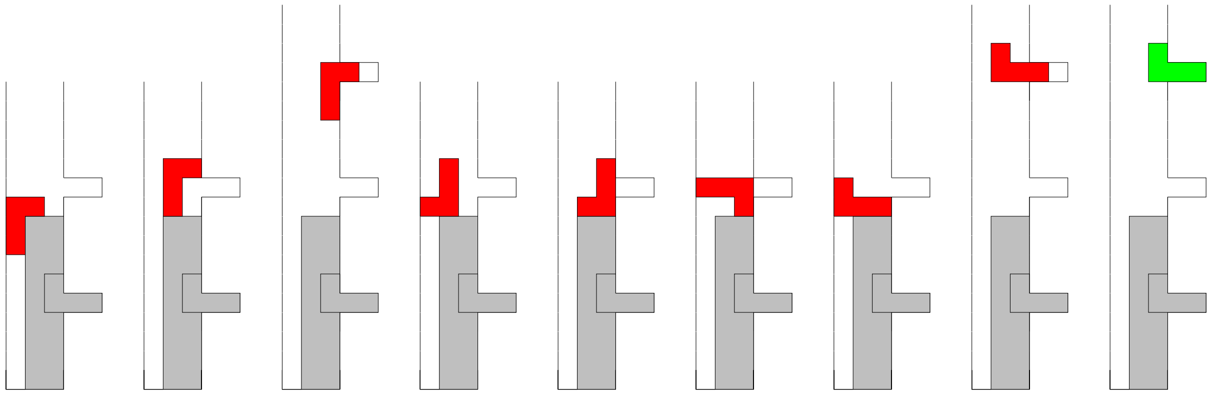
Proof. The possible configurations are the following:



Both have holes. □

Proposition C.15 *If* LG *is placed validly in a tall-plateau bucket, the result is a (*LG*-TP-i) for some i.*

Proof. The following configurations are possible:

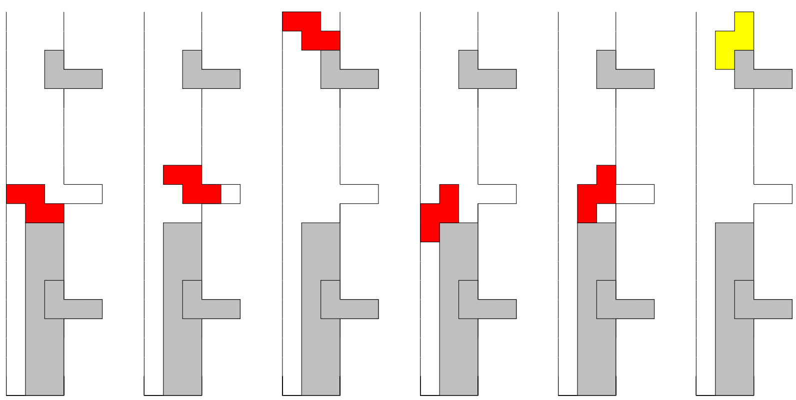


(In the third configuration, the LG is placed in the ith-highest bucket, for $i > 1$, and in the eighth and ninth, for $i \geq 1$.)

Of these, there is a hole in all but the last. The last is a (LG-TP-i). □

Proposition C.16 *There is no valid placement of* LS *in (*LG*-TP-i) for any i.*

Proof. The following configurations are possible:



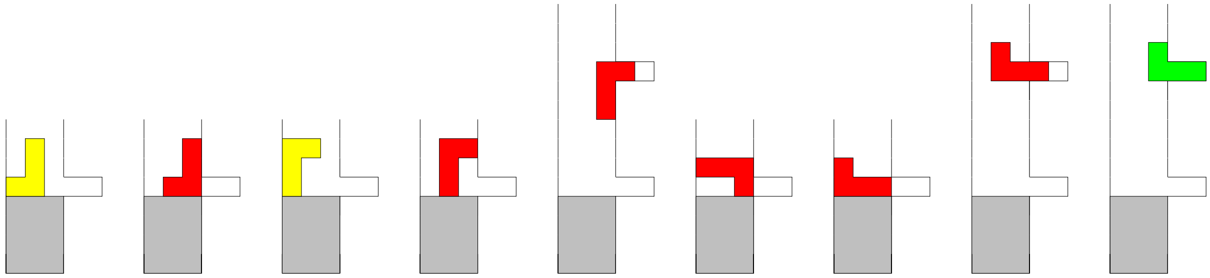
(Initially the ith-highest notch is filled by the LG. The second configuration denotes the LS in the jth-highest notch, for any $j \neq i$. The fifth configuration is blocked for $i = 1$.)

Of these, there is a hole in all but the last, which has a balconied flat-bottomed 2/3-ceiling discrepancy (or a balconied 2-veranda if $i = 1$). □

C.1.6 Underflat Buckets

Proposition C.17 *If* LG *is dropped into an underflat bucket, it produces (*LG*-UF-i) for some i.*

Proof. The possible configurations are the following:

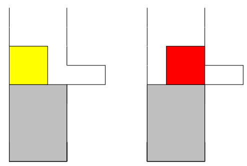


(The fifth of these applies to the ith-highest notch for any $i > 1$, the eighth and ninth for any $i \geq 1$.)

The first configuration creates a spurned notch, and the third has a balconied 111-floored notch. Of the others, all but the last—(LG-UF-i)—create holes. □

Proposition C.18 *There is no valid move for* Sq *in an underflat bucket.*

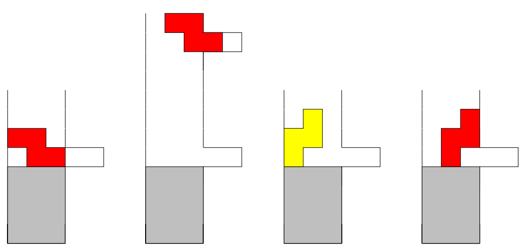
Proof. The possible configurations are the following:



The first of these configurations has a spurned notch, and the second has a hole. □

Proposition C.19 *There is no valid move for* LS *in an underflat bucket.*

Proof. The possible configurations are the following:

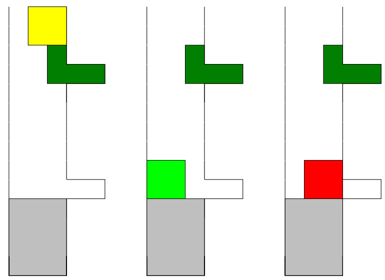


(The second has the LS in the ith-highest notch, for any $i \geq 1$.)

The first, second, and fourth of these configurations have holes; the second has a balconied 111-floored notch. □

Proposition C.20 *If* Sq *is dropped validly into (*LG-*UF-i), then (1) $i = 1$, and (2) the result is an overflat bucket.*

Proof. The possible configurations are as follows:



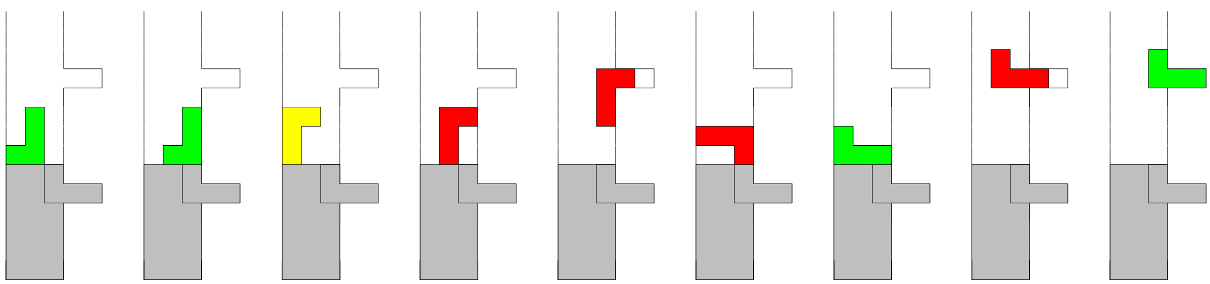
(In each of these configurations, the ith-highest notch is initially filled with the LG, for any $i \geq 1$. The third configuration is impossible if $i = 1$.)

The first of these is a balconied 111-floored notch for $i > 1$ and a balconied 2-veranda for $i = 1$. The second is a spurned notch unless the lowest notch is filled, i.e., unless $i = 1$; in this case, the reuslt is overflat. The third configuration has a hole. □

C.1.7 Overflat Buckets

Proposition C.21 *If* LG *is validly placed in an overflat bucket, the result is (*LG*-OF-1), (*LG*-OF-2), (*LG*-OF-3), or (*LG*-OF-4-i*) for some* i*.*

Proof. The possible configurations for placement of the LG are as follows:

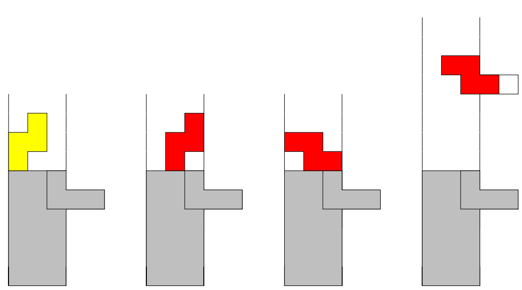


(The fifth, eight, and ninth denote the placement of the LG in the ith-highest notch, for any $i \geq 1$.)

The first two configurations are (LG-OF-1) and (LG-OF-2). The third has a balconied 2-veranda in the second column; the fourth, fifth (for any i), sixth and eighth (for any i) have holes. The seventh is (LG-OF-3), and the last is (LG-OF-4-i). □

Proposition C.22 *There is no valid move for* LS *in an overflat bucket.*

Proof. The possible configurations are:

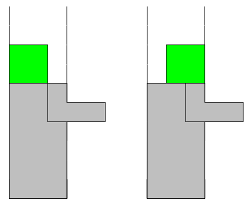


(The fourth represents the placement of the LS into the ith notch for any $i \geq 1$.)

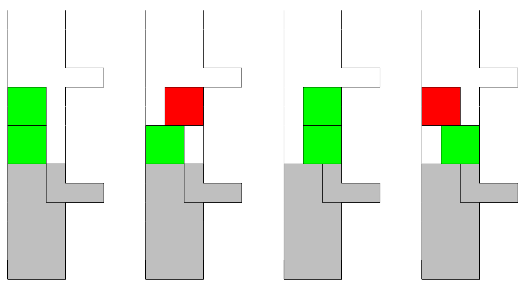
The first of these configurations has a balconied 1-veranda in non-notch rows; the remaining three have holes. □

Proposition C.23 *If* $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$ *is validly placed in an overflat bucket, then the result is an unprepped bucket.*

Proof. The possible configurations after the placement of the first Sq are as follows:



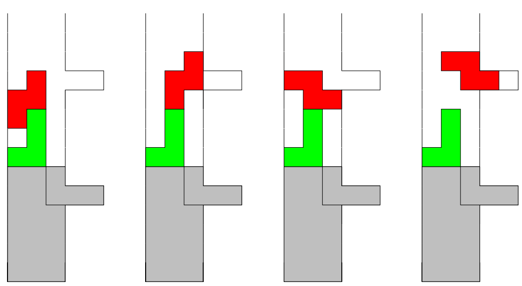
Both of these configurations are valid. Now, when we place the second Sq, the possible results are the following:



The second and fourth configurations have holes; the first and third are both unprepped. □

Proposition C.24 *There is no valid move for* LS *in (*LG*-OF-1).*

Proof. The only possible configurations are as follows:

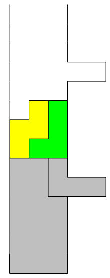


(The fourth denotes the placement of the LS into the ith-highest notch, for any $i \geq 1$.)

All of these configurations have holes. □

Proposition C.25 *There is no valid placement of* LS *into (*LG*-OF-2).*

Proof. Then the possible configurations are the following:

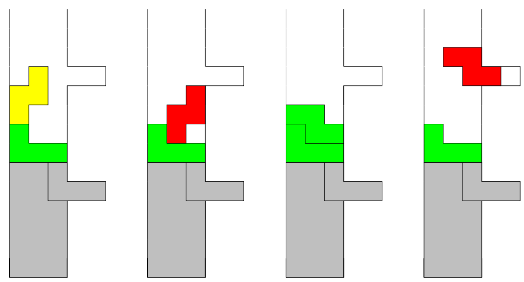


(The fourth denotes the placement of the LS into the ith-highest notch, for any $i \geq 1$.)

The first configuration is unapproachable, and the other three have holes. □

Proposition C.26 *If* LS *is placed validly in (*LG*-OF-3), the result is a trigger-happy bucket.*

Proof. The only possible configurations are the following:

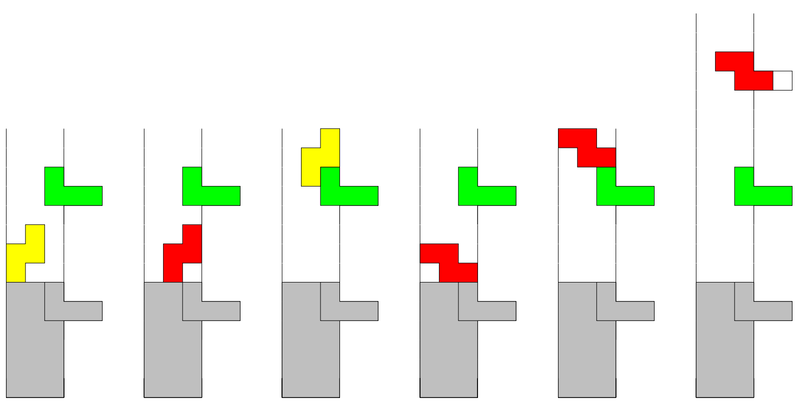


(The fourth denotes the placement of the LS into the ith-highest notch, for any $i \geq 1$.)

The first has a spurned notch, and the second and fourth have holes. The third is trigger-happy, as desired. □

Proposition C.27 *There is no valid move for* LS *in (*LG*-OF-4-i) for any i.*

Proof. Then the possible configurations are the following:



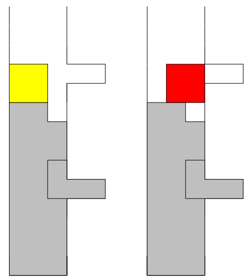
(Initially, in each configuration, there is an LG in the ith-highest notch, for some $i \geq 1$. The sixth configuration denotes the placement of the LS into the jth-highest notch, for any $j \neq i$ and $j \geq 1$.)

The first has a balconied 1-veranda in non-notch rows. The second, fourth, fifth, and sixth (for any j) have holes, and the third has a balconied flat-bottomed 2/3-ceiling discrepancy. □

C.1.8 Trigger-Happy Buckets

Proposition C.28 *There is no valid move for* Sq *in a trigger-happy bucket.*

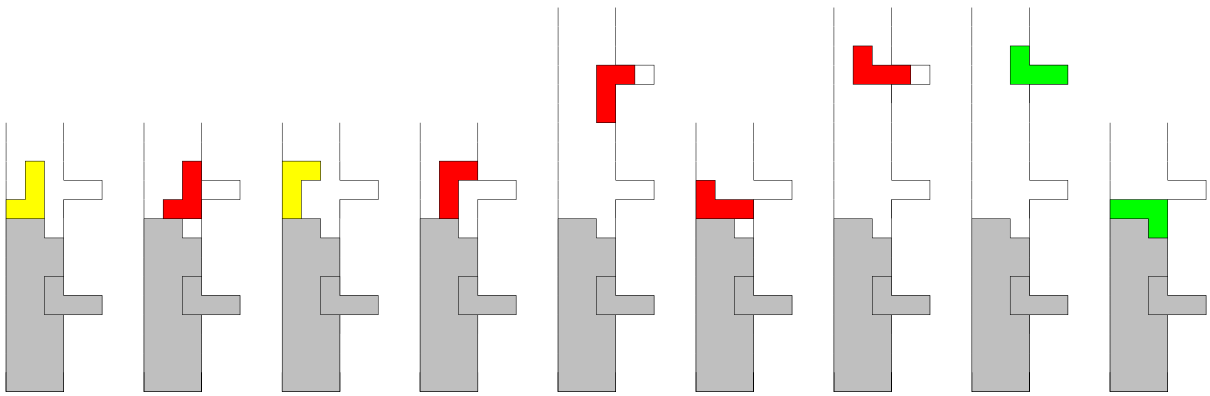
Proof. The possible placements for a Sq in a trigger-happy bucket are as follows:



The first has a spurned notch; the second has a hole. ☐

Proposition C.29 *If* LG *is placed validly into a trigger-happy bucket, the resulting configuration is either (*LG*-TH-i) or underflat.*

Proof. The possible configurations for the LG in the trigger-happy bucket are as follows:

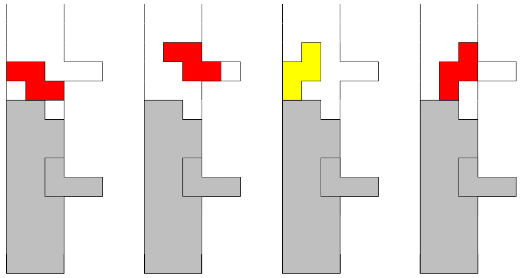


(In the fifth, seventh, and eighth configurations, the LG is in the ith-highest notch for any $i \geq 1$.)

The first has a spurned notch, and the third has a balconied 2-veranda. The eighth is (LG-TH-i), and the ninth is underflat. The remainder have holes. ☐

Proposition C.30 *There is no valid move for* LS *in a trigger-happy bucket.*

Proof. The possible configurations are as follows:

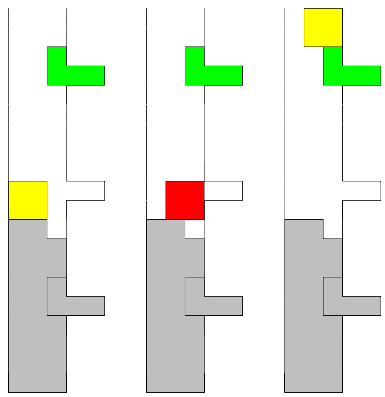


(The second configuration represents the placement of the LS into the ith-highest notch for any $i \geq 1$.)

The first, second (for any i), and fourth configurations have holes; the third has a spurned notch. □

Proposition C.31 *There is no valid placement of* Sq *in (*LG*-TH-i).*

Proof. The possible configurations are the following:

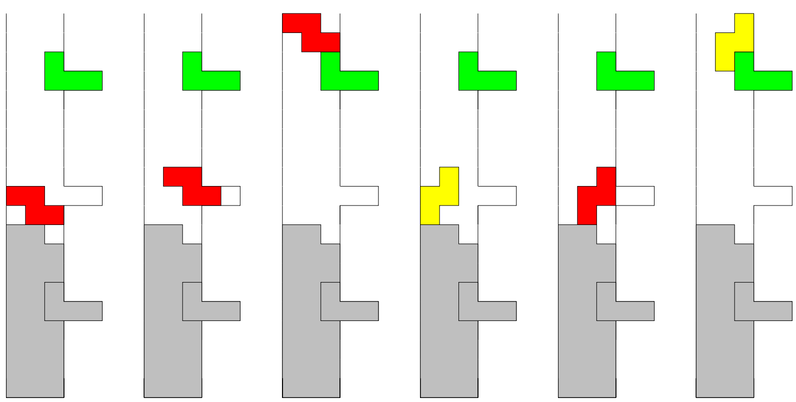


(The LG is initially in the ith-highest notch for $i \geq 1$; the second configuration is blocked for $i = 1$.)

The first has a spurned notch for $i > 1$ and a hole for $i = 1$. The second has a hole. The third has a balconied unfilled sub-notch rectangle if $i > 1$ and a balconied 2-veranda if $i = 1$. □

Proposition C.32 *There is no valid placement of* LS *in (*LG*-TH-i).*

Proof. The possible configurations are the following:

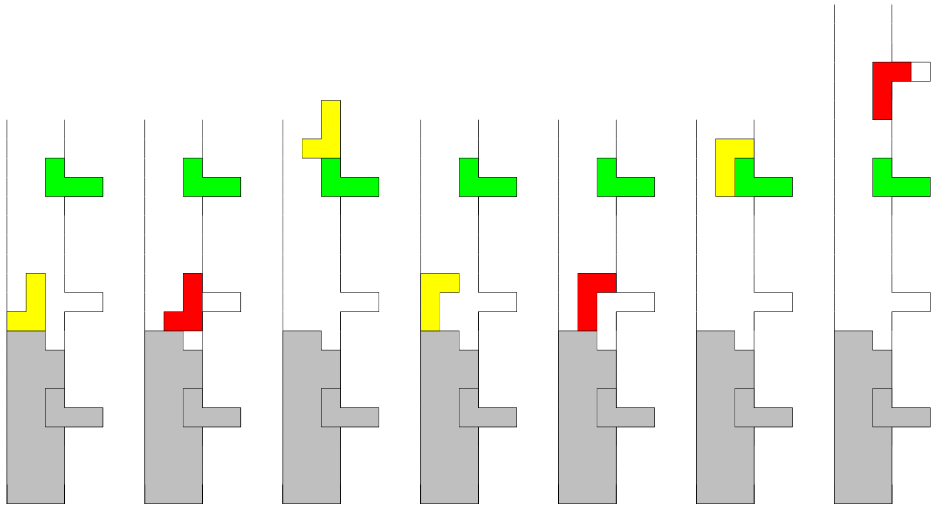


(The LG is initially in the ith-highest notch for $i \geq 1$; the fifth configuration is blocked for $i = 1$. In the second configuration, the LS is in the jth-highest notch for any $j \neq i$ and $j \geq 1$.)

The fourth configuration has a spurned notch for $i > 1$, and a hole for $i = 1$. The sixth has a balconied unfilled sub-notch rectangle for $i > 1$ and a balconied 2-veranda for $i = 1$. All other configurations have holes. □

Proposition C.33 *If* LG *is placed validly into (*LG*-TH-i) then the result is either (*LG*-UF-i) or (*LG*-TH-$\{i, j\}$).*

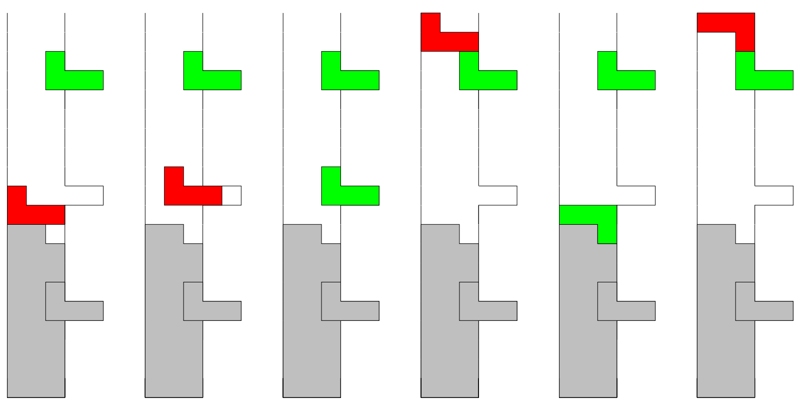
Proof. The possible configurations are as follows when the LG is placed vertically:



(The initially-placed LG is in the ith-highest notch, for some $i \geq 1$. The second and fifth configurations are blocked for $i = 1$. In the last configuration, the LG is in the jth-highest notch, for any $j \neq i$ and $j \geq 1$.)

The first configuration has a spurned notch for $i > 1$ and a hole for $i = 1$. The second and fifth have a hole. The third and sixth each have a balconied unfilled sub-notch rectangle for $i > 1$ and a balconied 2-veranda for $i = 1$. The fourth has a balconied 2-veranda for $i > 1$ and a hole for $i = 1$. The last has a hole for any j.

The possible configurations are as follows when the LG is placed horizontally:

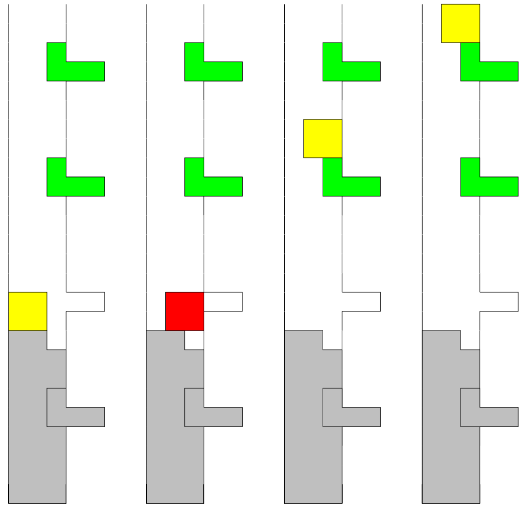


(The initially-placed LG is in the ith-highest notch for some $i \geq 1$; in the second and third configurations, the LG is placed into the jth-highest notch, for $j \neq i$ and $j \geq 1$.)

The third configuration is (LG-TH-$\{i, j\}$); the fifth is (LG-UF-i). The remainder all have holes. □

Proposition C.34 *There is no valid move for* Sq *in (*LG-*TH*-$\{i, j\}$*).*

Proof. The possible configurations are the following:



(The initial LG's are in the ith- and jth-highest notches, for some $i, j \geq 1$. The second configuration is blocked if $\min(i, j) = 1$.)

The first has a spurned notch (or a hole if $\min(i, j) = 1$), the second has a hole, and the third and fourth both have a balconied unfilled sub-notch rectangle (or a balconied 2-veranda if $\min(i, j) = 1$). $\square$

C.2 Soundness Theorems

Lemma 5.11 *In an unprepped configuration, the only possibly valid strategy for* $\langle \mathsf{I}, \mathsf{LG}, \mathsf{Sq} \rangle$ *is to place all three pieces in some bucket to produce an overflat bucket, yielding a one-overflat configuration.*

Proof. Initially, all buckets are unprepped. By Proposition C.1, the result of placing the I in an unprepped bucket is either underflat or (I-UP).

- **I produces an underflat bucket.** Then the configuration is one-underflat, and consists of unprepped buckets and an underflat bucket.
  - **The LG goes into the underflat bucket.** By Proposition C.17, the result is (LG-UF-i) for some i.

    Then the current configuration consists of unprepped buckets and (LG-UF-i). Now we consider where we can place the Sq:
    - **The Sq goes into the (LG-UF-i) bucket.** By Proposition C.20, this is valid iff $i = 1$ and the result is overflat.
    - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
  - **The LG goes into an unprepped bucket.** By Proposition C.2 the result is (LG-UP-i) for some i. Then the current configuration consists of unprepped buckets, an underflat bucket, and (LG-UP-i). But now we must place the Sq:
    - **The Sq goes into the (LG-UP-i) bucket.** Invalid by Proposition C.8.
    - **The Sq goes into the underflat bucket.** Invalid by Proposition C.18.
    - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.

  Thus the only valid move sequence is to place the LG in the same bucket to yield a (LG-UP-1), and then the Sq in the same bucket to yield an overflat bucket.

- **I produces an (I-UP) bucket.** Then the current configuration consists of unprepped buckets and an (I-UP) bucket.
  - **The LG goes into the (I-UP) bucket.** By Proposition C.5, the result is (I-UP-LG-i) for some i.

    Then the current configuration consists of unprepped buckets and (I-UP-LG-i).
    - **The Sq goes into the (I-UP-LG-i) bucket.** Invalid by Proposition C.7.
    - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
  - **The LG goes into an unprepped bucket.** By Proposition C.2 the result is (LG-UP-i) for some i. Then the current configuration consists of unprepped buckets, (LG-UP-i), and (I-UP). But now we must place the Sq:
    - **The Sq goes into the (LG-UP-i) bucket.** Invalid by Proposition C.8.
    - **The Sq goes into the (I-UP) bucket.** Invalid by Proposition C.6.
    - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.

Thus the only valid move sequence is to place the I into a bucket to yield an underflat configuration, then place the LG in the same bucket to yield a (LG-UP-1), and finally the Sq in the same bucket to yield an overflat. □

Lemma C.35 *In a one-overflat configuration, the only possibly valid strategy for the sequence* $\langle \mathsf{LG}, \mathsf{LS} \rangle$ *is to place both pieces into the overflat bucket, producing a trigger-happy bucket, yielding a one-trigger-happy configuration.*

Proof. The initial configuration consists of unprepped buckets and one overflat bucket.

- **The LG goes into an unprepped bucket.** By Proposition C.2, the result is (LG-UP-i) for some i. Then the configuration now consists of unprepped buckets, one overflat bucket, and one (LG-UP-i) bucket.
  - **The LS goes into the overflat bucket.** Invalid by Proposition C.22.
  - **The LS goes into the (LG-UP-i) bucket.** Invalid by Proposition C.9.
  - **The LS goes into an unprepped bucket.** Invalid by Proposition C.4.
- **The LG goes into the overflat bucket.** By Proposition C.21, the result is (LG-OF-1), (LG-OF-2), (LG-OF-3), or (LG-OF-4-i) for some i. Then the current configuration consists of unprepped buckets and one (LG-OF-1), (LG-OF-2), (LG-OF-3), or (LG-OF-4-i).
  - **The LS goes into an unprepped bucket.** Invalid by Proposition C.4.
  - **The LS goes into the (LG-OF-1) bucket.** Invalid by Proposition C.24.
  - **The LS goes into the (LG-OF-2) bucket.** By Invalid by Proposition C.25.
  - **The LS goes into the (LG-OF-3) bucket.** By Proposition C.26, the result is trigger-happy.
  - **The LS goes into the (LG-OF-4-i) bucket.** Invalid by Proposition C.27.

Thus the only possibly valid move is the placement of the LG into the overflat bucket to yield either a (LG-OF-2) or a (LG-OF-3), and the placement of the LS into the same bucket to make it trigger-happy. This results in a one-trigger-happy configuration. □

Lemma C.36 *In a one-trigger-happy configuration, the only possibly valid strategy for the sequence* $\langle \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ *is (1) to place all three pieces in the trigger-happy bucket to yield a one-overflat configuration, or (2) to place all three pieces in an unprepped bucket to yield a one-tall-plateau-one-trigger-happy configuration.*

Proof. Our initial configuration consists of unprepped buckets and one trigger-happy bucket. First, we consider whether zero, one, or two of the LG's go into the trigger-happy bucket:

- **Neither LG goes into the trigger-happy bucket.** Then the two LG's go into unprepped buckets.
  - **The two LG's go into different unprepped buckets.** Then by Proposition C.2, the result of dropping each LG is (LG-UP-i) and (LG-UP-i'), for some i and i'.

    Thus our configuration consists of unprepped buckets, (LG-UP-i), (LG-UP-i'), and a trigger-happy bucket.
    * **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
    * **The Sq goes into (LG-UP-i) or (LG-UP-i').** Invalid by Proposition C.8.
    * **The Sq goes into the trigger-happy bucket.** Invalid by Proposition C.28.

- **The two LG's go into the same unprepped bucket.** Then by Proposition C.2, the result of dropping the first LG into an unprepped bucket is (LG-UP-i) for some i. By Proposition C.10, the result of the second is a short-plateau or (LG-UP-$\{i,j\}$), for some j.

  Now our configuration consists of unprepped buckets, a trigger-happy bucket, and one (LG-UP-$\{i,j\}$) or short-plateau.

  * **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
  * **The Sq goes into the trigger-happy bucket.** Invalid by Proposition C.28.
  * **The Sq goes into (LG-UP-$\{i,j\}$).** Invalid by Proposition C.11.
  * **The Sq goes into the short-plateau bucket.** By Proposition C.12, the result is a tall-plateau.

- **Exactly one of the LG's goes into the trigger-happy bucket.** By Proposition C.29, the result of placing the LG into the trigger-happy bucket is either (LG-TH-i) or underflat, and by Proposition C.2, the result of dropping the other LG into an unprepped column is (LG-UP-i'), for some i and i'.

  The resulting configuration then consists of unprepped buckets, (LG-UP-i') and either (LG-TH-i) or underflat.

  - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
  - **The Sq goes into (LG-UP-i').** Invalid by Proposition C.8.
  - **The Sq goes into (LG-TH-i).** Invalid by Proposition C.31.
  - **The Sq goes into underflat.** Invalid by Proposition C.18.

- **Both of the LG's go into the trigger-happy bucket.** By Proposition C.29, the result of placing the LG into the trigger-happy bucket is either underflat or (LG-TH-i) for some i.

  By Proposition C.33, the result of placing the second LG in (LG-TH-i) is either (LG-UF-i) or (LG-TH-$\{i,j\}$) for some j. By Proposition C.17, the result of placing the LG in the underflat bucket is also (LG-UF-i).

  Then, regardless of whether the first LG produced (LG-TH-i) or underflat, the current configuration consists of unprepped buckets and either (LG-UF-i) or (LG-TH-$\{i,j\}$).

  - **The Sq goes into an unprepped bucket.** Invalid by Proposition C.3.
  - **The Sq goes into (LG-UF-i).** By Proposition C.20, this is valid iff $i = 1$ and the result is overflat.
  - **The Sq goes into (LG-TH-$\{i,j\}$).** Invalid by Proposition C.34.

Thus the only possibly-valid moves are to place all three pieces into (1) the trigger-happy bucket to produce an overflat bucket, yielding a one-overflat configuration, or (2) an unprepped bucket to produce a tall-plateau bucket, yielding a one-tall-plateau-one-trigger-happy configuration. $\square$

Lemma C.37 *There is no valid move for the sequence $\langle \mathsf{LG}, \mathsf{LS} \rangle$ in a one-tall-plateau-one-trigger-happy configuration.*

Proof. Our initial configuration consists of unprepped buckets, one tall-plateau bucket, and one trigger-happy bucket.

- **The LG goes into an unprepped bucket.** Then by Proposition C.2, the result is (LG-UP-i) for some i.
  - **The LS goes into an unprepped bucket.** Invalid by Proposition C.4.
  - **The LS goes into (LG-UP-i).** Invalid by Proposition C.9.
  - **The LS goes into the tall-plateau bucket.** Invalid by Proposition C.13.
  - **The LS goes into the trigger-happy bucket.** Invalid by Proposition C.30.
- **The LG goes into the tall-plateau bucket.** Then by Proposition C.15, the result is (LG-TP-i) for some i.
  - **The LS goes into an unprepped bucket.** Invalid by Proposition C.4.
  - **The LS goes into (LG-TP-i).** Invalid by Proposition C.16.
  - **The LS goes into the trigger-happy bucket.** Invalid by Proposition C.30.
- **The LG goes into the trigger-happy bucket.** Then by Proposition C.29, the result is either underflat or (LG-TH-i) for some i.
  - **The LS goes into an unprepped bucket.** Invalid by Proposition C.4.
  - **The LS goes into (LG-TH-i).** Invalid by Proposition C.32.
  - **The LS goes into the underflat bucket.** Invalid by Proposition C.19.
  - **The LS goes into the tall-plateau bucket.** Invalid by Proposition C.13.

Thus there is no valid strategy for $\langle \mathsf{LG}, \mathsf{LS} \rangle$ in a one-tall-plateau-one-trigger-happy configuration. □

Lemma 5.12 *For the sequence $\langle \mathsf{LG}, \mathsf{LS}, \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$:*

1. *In a one-overflat configuration, the only possibly valid strategy is either (1) to place all pieces in the overflat bucket to produce an overflat bucket, yielding a one-overflat configuration, or (2) to place $\langle \mathsf{LG}, \mathsf{LS} \rangle$ into the overflat bucket and $\langle \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ in an unprepped bucket, yielding a one-tall-plateau-one-trigger-happy configuration.*
2. *In a one-tall-plateau-one-trigger-happy configuration, there is no valid strategy.*

Proof. For the one-overflat configuration, by Lemma C.35, if the sequence $\langle \mathsf{LG}, \mathsf{LS} \rangle$ is placed validly, both pieces are dropped into the overflat bucket, and the result is a one-trigger-happy configuration. By Lemma C.36, the only valid placement for the sequence $\langle \mathsf{LG}, \mathsf{LG}, \mathsf{Sq} \rangle$ in a one-trigger-happy configuration yields a one-overflat (placing all three pieces into the trigger-happy bucket) or one-tall-plateau-one-trigger-happy configuration (placing them into an unprepped bucket).

In the one-tall-plateau-one-trigger-happy configuration, by Lemma C.37, there is no valid trajectory sequence. □

Lemma 5.13 *For the sequence $\langle \mathsf{Sq}, \mathsf{Sq} \rangle$,*

1. *In a one-overflat configuration, the only possibly valid strategy is to place both pieces in the overflat bucket to produce an unprepped bucket, yielding an unprepped configuration.*

2. *In a one-tall-plateau-one-trigger-happy configuration, there is no valid strategy.*

Proof. By Propositions C.3, C.28, and C.14, no Sq can validly go into any unprepped, trigger-happy, or tall-plateau bucket.

For a one-overflat configuration, then, both Sq's must go into the overflat bucket. By Proposition C.23, the result is an unprepped bucket.

For a one-tall-plateau-one-trigger-happy configuration, there is no bucket into which the first Sq can be validly placed. □