Tetris and Decidability ⋆

Hendrik Jan Hoogeboom, Walter A. Kosters

Leiden Institute of Advanced Computer Science (LIACS)
Universiteit Leiden, The Netherlands

Abstract

We consider a variant of TETRIS where the sequence of pieces (together with their orientation and horizontal position, which cannot be changed anymore) is generated by a finite state automaton. We show that it is undecidable, given such an automaton, and starting from an empty game board, whether one of the generated sequences leaves an empty game board. This is contrasted with more common situations where piece translations and rotations are allowed.

Key words: Tetris, Decidability, Theory of computation.

1 Introduction

The discovery that the well-known game of TETRIS is provably hard was noticed in the popular press. To be precise, Demaine, Hohenberger and Liben-Nowell show that the problem of deciding whether a given TETRIS configuration can be cleared using a given sequence of pieces is NP-hard [4,1]. Thus, this game gives rise to interesting combinatorial problems, extending those of classical tilings [5].

Here we present another fundamental result on TETRIS, albeit for a restricted version of the game, where no user intervention is possible. As in the standard (normal) game (see [4]) the sequence of pieces that are presented to the player is generated by a random process. Formulated otherwise, we assume that this process is finite state: the sequence of consecutive pieces together with their orientation and initial position on top of the board is randomly chosen from a given regular language describing such sequences. Then, the game proceeds deterministically, as if the buttons that enable the player to rotate and translate the pieces are malfunctioning; each piece just falls down into the configuration from its starting position in its initial orientation.

For any given sequence it is clearly decidable whether this leaves an empty board or not (even with user intervention): just try the finitely many possibilities. This leaves open the *existential* question, whether a given set of sequences contains a sequence that leaves a completely cleared game board (starting from an initially empty game board). We show the undecidability of that question, where the set is a *regular* language of TETRIS pieces with their initial orientation and position.

It is remarkable that we only need copies of a

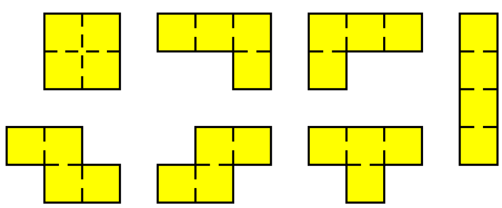
⋆ Information Processing Letters 89 (2004), 267-272.

Email addresses: hoogeboom@liacs.nl (Hendrik Jan Hoogeboom), kosters@liacs.nl (Walter A. Kosters).

single TETRIS piece, the dash or `I`, to obtain this result. We contrast this by showing that *with* user intervention, and optimal play, the sequences consisting of either pieces of one single kind *or* dashes and squares that leave an empty game board have a simple property that makes the problem decidable.
Most (un)decidability results we present here are valid for boards of width 10, which happens to be the standard size.

2 The Game

The game of TETRIS is played on a rectangular board consisting of cells. The board is of fixed width and, for our purposes, of unbounded height. Seven game pieces can be used, each covering four board cells; they are depicted below. These pieces are known as (top row, left to right) `O` or *square*, `J` or *leftgun*, `L` or *rightgun*, `I` or *dash*; and (bottom row) `Z` or *leftsnake*, `S` or *rightsnake*, and `T` or *tee*:



The computer generates a sequence of pieces that drop down from the top of the board until they rest on top of previously dropped pieces or on the bottom of the game board. The user can determine the position and orientation of the pieces by rotating and moving them horizontally while they fall. Whenever all the cells of a row of the game board are occupied, the line is cleared; all occupied cells above it are lowered by one row (and no more). This row clearing can happen for several lines simultaneously.
In TETRIS the purpose usually is to clear as many rows as possible given the generated sequence of pieces, while avoiding running out of space vertically. As the game of TETRIS itself is finite state (and hence decidable) when played on a board of given width and height, here we assume the board is of unbounded height.

We now consider different *models of user intervention*. On the one hand we have the normal TETRIS rules, as described above, where the user has many possibilities to influence the result. At the other extreme we have the model where the user is not allowed any intervention: once the computer fixes the piece, its orientation and its horizontal position, the piece drops down in the specified orientation, and in the specified position.

As for a given game board the number of initial possibilities of each piece — its orientation and the columns occupied — is bounded, the sequence of pieces dropped can be described by a string over a finite alphabet.

This suggests the following decision problem, TETRIS *with Intervention Model* M:

Instance. A regular language L describing sequences of TETRIS pieces (with their initial orientation and horizontal position) for a given width game board.

Question. Is there a sequence in L that leaves the game board empty after dropping all the pieces into an initially empty game board, according to the model M? I.e., does the user have a way to clear the entire sequence, while adhering to the rules in M?

Note that if the user is not allowed any intervention (we call this the *Null Intervention Model*, and refer to the corresponding decision problem as TETRIS *without intervention*), there are no choices to be made. For more complicated models, we are looking for "optimal" user actions that lead to total clearings.

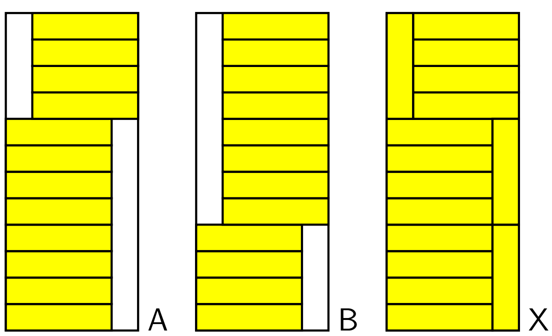
3 No Intervention

In this section we shall examine the decision problem for the Null Intervention Model. We then have the following undecidability result:

Theorem 1 TETRIS *without intervention, for sequences consisting only of* `I`*'s on a board of width 10, is undecidable.*

PROOF. We show the undecidability by reduction from the Post Correspondence Problem [8]. Given an instance of the PCP — two sequences $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$ of strings over a two-letter alphabet $\{a, b\}$ — we construct an instance L of TETRIS without intervention, on a board of width 10. The left and right halves of the board (each a board of 5 cells wide) will act as stacks holding proposed solutions to the PCP, i.e., words of the form $u_{i_1} u_{i_2} \ldots u_{i_k} = v_{i_1} v_{i_2} \ldots v_{i_k}$ for some $k \geq 1$, and $1 \leq i_1, i_2, \ldots, i_k \leq n$.

To build the contents of the stacks we need three basic blocks, that we call A, B and X. The first two of these represent the two symbols a and b of the alphabet of the PCP; the last one is a block used for padding the two copies of the solutions:



Note that the A and B blocks can be removed (popped from the stack) following the rules of TETRIS, by dropping three vertical `I`'s in the proper columns, provided the blocks are next to an X block on the other stack. The blocks are designed in such a way that the vertical `I`'s used to remove the blocks do not fall through to the next block. Pieces dropped in the first column are blocked by the bottom rows of the block, pieces falling in the fifth column are blocked by the topmost row of the block below (or by the floor).

First, the language L (or the corresponding finite automaton) prepares nondeterministically a sequence of blocks pushing onto the two stacks the same (nonempty) sequence of A's and B's, but randomly interleaved with X's. This part is independent of the particular instance of the PCP.

Then, in a second phase, L tries to clear the board, guessing a solution of the PCP, by repeatedly picking an index i ($1 \leq i \leq n$) and trying to pop the left stack according to the string u_i and the right stack according to the string v_i.

We show the equivalence of the PCP and its implementation as a TETRIS-problem in two implications.

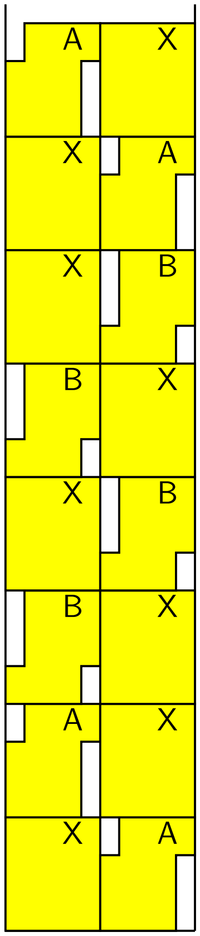
(1) If the particular PCP has a solution, then the language L has a way to leave the empty game board.

In the first phase L may leave (in a non-deterministic manner) a game board according to a solution $u_{i_1} u_{i_2} \ldots u_{i_k} = v_{i_1} v_{i_2} \ldots v_{i_k}$ of the PCP such that from top to bottom alternately the left stack is filled with A and B blocks representing u_i and the right stack is filled with the same number of X's, and the right stack is filled with A and B blocks representing v_i and the left stack is filled with the same number of X's.

In this way, a "perfect" configuration as in the figure below is obtained, which obviously can be cleared in the second phase according to the simulation of a solution of the PCP.

(2) If the language L has a way to leave the empty game board, then the original PCP has a solution.

Popping the stacks obviously corresponds to verifying the guessed solution of the PCP, so we only have to check that the process of popping blocks is well behaved: trying to pop top-

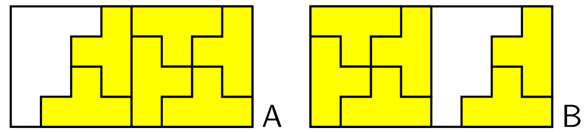


A "perfect" configuration left after the first phase of our construction: in the left stack we can read (top-to-bottom) a, b, ba while we encounter ab, b, a in the right stack.

most A cannot lead to a cleared game board unless this is done while the topmost block indeed equals A, and similarly for B.
Popping an A involves dropping an `I` in the first column of the stack, and two `I`'s in the fifth column of the stack. Applied to topmost B or X this leads to at least one vertical `I` that remains on top of the stack in the fifth column. This piece can never be removed as we do not drop any pieces covering the second, third and fourth columns of the stack. Similarly we argue for trying to pop B while A or X is on top. □

The `I` piece considered in the undecidability result is one of the most simple ones among the TETRIS pieces. We now discuss decidability of TETRIS without intervention when the input is restricted to either one of the other pieces.

A simple argument shows that a nonempty sequence of either `S` or `Z` pieces cannot clear the board (cf. [3]), so the problem for those pieces becomes trivially decidable. For the pieces `T`, `L` and `J` we can conceive a configuration that can be used to construct stacks, and similar arguments as for `I` hold (albeit on a board of width 16). For example, for `T`:



Finally, for `O` only very regular patterns are possible that leave an empty board. This is the basis for the following result:

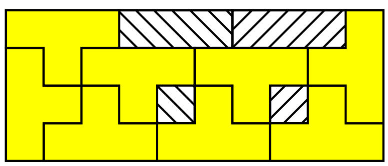
Theorem 2 TETRIS *without intervention, for sequences consisting only of* `O`*'s on a board of width 10, is decidable.*

PROOF. The only way to clear the bottom two rows is by filling them with adjacent `O`'s: putting two `O`'s a single cell apart makes it impossible for later `O`'s to fill the gap. The same holds for other rows. Hence sequences that clear the board are easily characterized: they should only drop `O`'s in first and second column, or third and fourth, or fifth and sixth, . . . , and have the same number of `O`'s in each of these positions. The order of the pieces is irrelevant as these pairs of columns can be filled independently. Hence, clearing sequences are characterized by the number of squares in each initial position, and we can dispose of their order.
This means that these sequences are given by a so-called semi-linear set [6], and it is effectively decidable whether the given regular language L has a nonempty intersection with this semi-linear set. □

4 Increasing Intervention

We reconsider the decision result of the last section, now allowing user translation and rotation of the pieces that are specified by the sequences in the given regular language L. The intervention is just as in the standard TETRIS game. We refer to the corresponding decision problem as TETRIS *with normal intervention*.

The general question is related to the many tiling problems for polyominoes (see, e.g., [7,5]), as the existence of a tiling of a rectangle by TETRIS tetrominoes implies a possible clearing of the board using the TETRIS pieces in some suitable order. However, apart from the fact that the TETRIS problem also deals with the *order* in which the pieces are offered, classical tiling is more restricted: it does not allow intermediate clearing of rows. As an example, ten T's can clear the TETRIS game board (of width 10, as below) whereas there is no tiling of the 10 by 4 rectangle using T's [9].



We restrict ourselves here to rectangular pieces. The sequences of the rectangular TETRIS pieces `O` and `I` that can be used to leave an empty game board have a simple characterization. Our result is valid for standard width 10, but can be stated slightly more generally.

Lemma 3 *A sequence of* `I`*'s and* `O`*'s can be dropped into an initially empty game board of width* $4k+2$ $(k \geq 1)$ *leaving the empty board if and only if*

(1) the number of pieces is a multiple of $2k+1$*, and*

(2) the number of `I`*'s is even.*

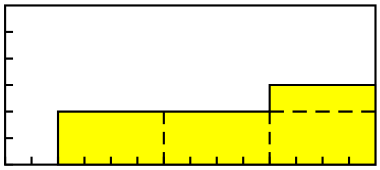
PROOF. First we argue that the conditions are *necessary*. Of course (1) follows from the fact that each line contains $4k+2$ cells whereas each piece covers 4 cells. With p pieces dropped and r rows cleared we must have $4p = r(4k+2)$, which implies that $2k+1$ must divide p. The same equation shows that the number of rows cleared must be even.

Regarding (2) we demonstrate that both the number of horizontal `I`'s and the number of vertical `I`'s must be even. For the vertical `I`'s we use a classical colouring argument. Colour the columns of the board alternately black and white. Each `O` and each horizontal `I` covers two black and two white cells. The vertical `I` however covers cells of a single colour, and the number of "white" and "black" vertical `I`'s must match.

Here in the TETRIS setting the symmetric argument (with rows coloured black and white) does not work to prove that the number of horizontal `I`'s is even, because we cannot guarantee that vertical `I`'s cover two black and two white cells in an alternating colouring of rows: clearing a row will disturb regularities in the chosen colouring.

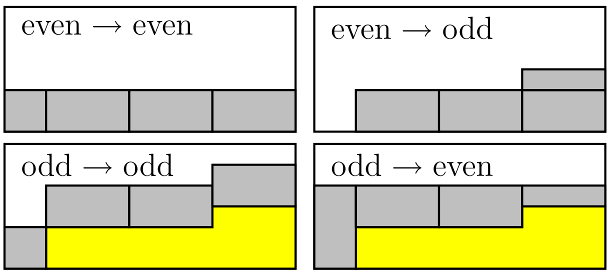
Note however, that every column has four cells for every vertical `I` it contains, two cells for every `O`, and a single cell for every horizontal `I` that was used to cover it. As every column contains an even number of occupied cells, that implies the column is touched by an even number of horizontal `I`'s. Now consider columns $1, 5, 9, \ldots, 4k+1$. Each horizontal `I` touches exactly one of these columns as they are four cells apart. This implies that the total number of horizontal `I`'s is again even.

We now show that the conditions are *sufficient*. Given a sequence of pieces satisfying (1) and (2) we drop them using a finite state strategy dealing with $2k+1$ pieces at a time. (This generalizes the customary game play of TETRIS, where the player has a lookahead of one piece.) The game board has two states *even* and *odd*, corresponding to the parity of the number of dropped `I`'s. When even, the board is empty (i.e., all lines cleared), when odd, the board is as follows, with $8k+4$ cells covered:



odd state, for $k = 3$

We switch between states using a sequence of $2k + 1$ pieces with an odd number of `I`'s, and we stay in a state otherwise. Here are the necessary constructions, where a large rectangle represents either two `O`'s or two `I`'s:



In the case of a single `I` and $2k$ `O`'s moving from the even state to the odd state (picture top right), we need an alternative construction where the `I` is *under* the `O`'s (i.e., is horizontally placed in the right hand corner) in the eventuality that `I` is the first or second piece. □

We have an immediate corollary:

Theorem 4 TETRIS *with normal intervention, for sequences consisting only of* `I`*'s and* `O`*'s on a board of width 10, is decidable.*

PROOF. The two requirements from the Lemma in fact describe a regular language. As regular languages are effectively closed under intersection, we can build the intersection of the given language L and the one specified by the requirements. This intersection can be tested for emptiness. □

Note that our construction forms in fact an ordinary tiling of the game board as no special use has been made of the TETRIS rules for clearing rows. It is close to the strategies developed for winning two piece TETRIS games presented in [2], except that here we want to leave an empty game board, whereas the strategies given in [2] only need to avoid filling too many rows of the board. Moreover, we allow a lookahead that differs from the standard single piece shown to the user.

The Lemma can be generalized to other widths. Without proof we state that for a game board of width $4k$, the following conditions characterize the sequences of `I`'s and `O`'s that leave the empty game board (under optimal play):

(1) the number of pieces is a multiple of k, and
(2) if this is an even multiple, then the number of `I`'s is even, and
(3) if this is an odd multiple, then the number of `I`'s is at least k, and the number of `I`'s minus k is even.

These conditions are slightly more complicated than the $4k + 2$ case above due to the fact that for width $4k$ the number of cleared rows can be odd (which can be obtained by dropping a row of k horizontal `I`'s). Also, for the sequence `O O O I I O` on a board of width 8, clearing the board essentially requires clearing a row while rows below it contain unoccupied cells.

Let us conclude with a slightly unexpected result. Restricted to a single piece (which can be other than the seven tetrominoes in standard TETRIS) TETRIS with normal intervention is decidable, even though we do not (need to) explicitly know the decision algorithm in each particular case.

Theorem 5 TETRIS *with normal intervention, for sequences consisting of copies a single fixed piece, on a board of fixed width, is decidable.*

PROOF. With a single piece the order of pieces is irrelevant, and we only need to consider their number. Note that if we can clear an initially empty board both with k_1 and k_2 pieces, then we can clear it with k_1+k_2 pieces.

If the board cannot be cleared with the given piece at all (as for `S` or `Z`, cf. [3]) then the problem is trivially decidable. Otherwise, let $K > 0$ be the least number of pieces that clears the board. For each $0 \le k < K$ there is a minimal number $v_k > 0$, $v_k \equiv k \bmod K$, such that v_k pieces clear the board (if there is no such value we take it to be infinite; and $v_0 = K$). Again, the sequences that clear the board are given by a semi-linear set (fixed by the v_k plus multiples of K) and we can decide if the given regular L has a nonempty intersection with this set. □

Observe that our proof above is not effective, as we have not provided a way to compute the values v_k given a single piece and the width of the board.
For `T` the number of pieces should be a multiple of 5. One can show that for a board of width 10, 15 pieces cannot clear the board, but 25 can. As we already noticed, 10 pieces can clear the board. So in this case we have the semi-linear set $\{10, 20, 25, 30, 35, \ldots\}$.

5 Conclusion

Even with only a single piece, TETRIS without intervention can be formulated as an "implementation" of the Post Correspondence Problem, and hence is undecidable. On the other hand, TETRIS with normal intervention is decidable for a single piece.
What remains open is the decidability question for TETRIS with normal intervention in general, and, when decidable, giving a characterization of the sequences that clear the board. We expect these to be difficult problems to tackle, even for restricted subsets of the seven TETRIS pieces, especially in cases where the strategy does not coincide with a tiling of the board in the classical sense (like our example for `T`). Note that also the *order* of the pieces plays a role, although this has not been very relevant in the (restricted) considerations of our previous section.

Acknowledgement. We are indebted to two anonymous referees for their helpful comments.

References

[1] R. Breukelaar, H.J. Hoogeboom and W.A. Kosters. Tetris is Hard, Made Easy. Technical Report, Leiden Institute of Advanced Computer Science, 2003.

[2] J. Brzustowski. Can You Win at Tetris? Master's Thesis, University of British Columbia, 1992.

[3] H. Burgiel. How to Lose at Tetris. Mathematical Gazette 81 (1997) 194–200.

[4] E.D. Demaine, S. Hohenberger and D. Liben-Nowell. Tetris is Hard, Even to Approximate. Computing and Combinatorics, 9th Annual International Conference, T. Warnow, B. Zhu (Eds.). Lecture Notes in Computer Science 2697, pages 351–363, 2003.

[5] B. Durand and L. Vuillon, editors, Tilings of the plane, Special issue of Theoretical Computer Science, Volume 303, Numbers 2–3, 2003.

[6] S. Ginsburg. *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, New York, 1966.

[7] S.W. Golomb. *Polyominoes: Puzzles, Patterns, Problems, and Packings*, 2nd ed., Princeton University Press, Princeton, NJ, 1995.

[8] E.L. Post. A Variant of a Recursively Unsolvable Problem. Bulletin of the American Mathematical Society 52 (1946) 264–268.

[9] D.W. Walkup. Covering a Rectangle with T-Tetrominoes. American Mathematical Monthly 72 (1965) 986–988.