2 Player Tetris is PSPACE Hard

Lev Reyzin*
Department of Computer Science, Yale University
lev.reyzin@yale.edu

Abstract

Tetris is a popular computer game in which players place falling tetrominoes on a board. While the complexity of single player has been previously studied, we analyze the complexity of the popular two player version of this game. We define the rules of two player tetris and show that under many reasonable assumptions 2 Player Tetris is PSPACE hard.

1 Introduction and Rules

Single player tetris is played on an m by n grid (or board). The player manipulates blocks, or tetrominoes, as they fall from the top of the board. When a row is completely filled by blocks, it disappears and the player scores points. The game continues until the player has no more room to place the falling blocks, at which point he loses. It has been shown that maximizing many different objective functions is NP-hard, even to approximate [1].

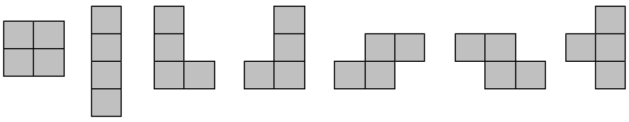

Figure 1: The possible tetronimoes in tetris.

Two player tetris is played on two m by n grids, with each player playing on his own board. Each grid begins in some initial configuration of occupied and free blocks. Tetrominoes fall from the top of each player's grid at constant speed. A player can rotate each tetromino by 90 degrees and move it left and right as it falls. Figure 1 shows the set of tetrominoes that can appear in the game. A tetromino becomes "placed" once a filled grid-point or the bottom of the board prevents it from falling further. After a tetromino is placed by a player on his board, any rows whose grid-points are completely occupied disappear from that player's board and the occupied grid points above the disappearing rows "fall" downward by as many rows as disappeared below them. We call this clearing a row. Then a new tetromino begins to fall.

When a player clears a row on his board, a configuration of blocks, which we will call the "penalty blocks," appears at the bottom of his opponent's board. The opponent's occupied blocks shift up by the number of rows appearing on his board. The configuration of the penalty blocks is a function of the last cleared rows that caused the blocks to appear. Sometimes the rules specify that the number of rows appearing must be 1 row fewer than the number of rows cleared. However, due to space limitations, in this extended abstract we show the result for the case that the penalty blocks are an arbitrary function of the last constant number of moves by both players. Yet the result extends to the stricter rules mentioned above. This rule allows players who are clearing rows to help make their opponents lose by taking away free grid points from them. A player wins the game when his opponent has a grid block filled on the upper-most row of his grid.

In this paper, we prove the following theorem.

Theorem 1. *2 Player Tetris (2PT) is PSPACE hard.*

2 Proof Sketch

The canonical PSPACE complete problem is TQBF, True Quantified Boolean Formula. A simple reduction from TQBF [3] shows that FORMULA-GAME is PSPACE-complete. FORMULA-GAME is a game played on a quantified boolean formula in prenex normal form

$$\phi = \exists x_1 \forall x_2 \exists x_3 ... \forall x_k [\psi]$$

where ψ is a boolean formula in CNF.

The game goes as follows: there are two players, E (for existential) and A (for universal). Players E (and A, respectively) set values for variables bound to the existential (and universal) quantifiers in the order they appear in ϕ. It is PSPACE hard to tell who has a winning strategy (E has a winning strategy if and only if ϕ is true).

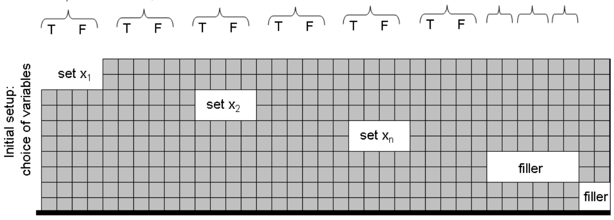

Figure 2: The starting position for player E

We will reduce the FORMULA-GAME to 2PT by showing how a game of 2PT can simulate the play of a FORMULA-GAME. This technique is similar to the technique used by Lichtenstein and Sipser to show Go is PSPACE hard [2].

This game of tetris will be played by two players, again player E and player A. The game begins with their boards

*Supported by a Yahoo! Research Kern Family Scholarship

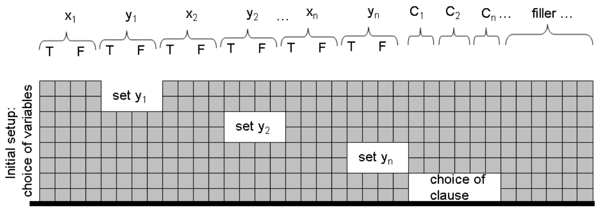

Figure 3: The starting position for player A

almost full such that they have no choice but to play through the game of setting variables in a QBF expression in a FORMULA-GAME or lose.

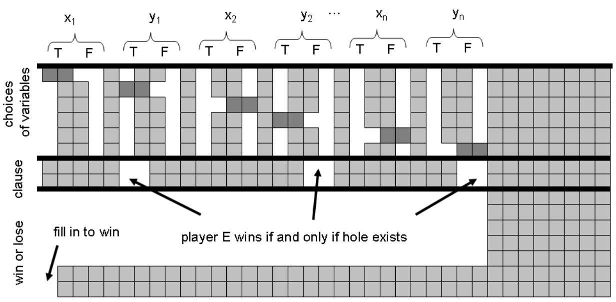

Figure 4: The widget appearing on player E's board that corresponds to the settings of the variables in the quantified boolean formula being played. This widget has x_1 set to *false*, y_1 set to *false*, x_2 set to *true*, etc.

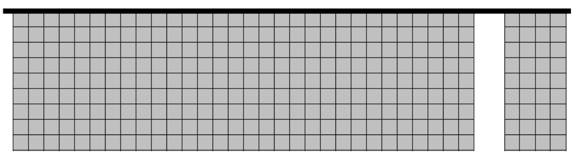

Figure 5: The filler appearing on player A's board

In Figure 2, we present the initial configuration for player E. The labeled empty blocks in E's initial configuration correspond to player E's choices of the settings of the variables. In Figure 3, we present the initial configuration for player A's board. As player A and E play tetris against each other, they make choices about the variable settings in the manner they clear their rows. As they make these choices, as player A clears rows, the widget in Figure 4 begins to appear on E's board. If either player deviates from this prescribed play, the deviating player immediately loses the game. The final widget that comes into view on player E's board either wins him the game (if the QBF is true) or loses him the game (if the formula is false).

One important aspect of this proof is the idea of timing. We would like the players to alternate between moving their pieces. If we allow "dropping" pieces quickly, we could assume that both players play at the same speed. Or that if they both played "optimally" they would both play equally quickly. Or, we could not allow players to "drop" pieces, in which case we could make the pieces fall at the same rates on both boards. Since through their game, both boards are (equally) filled almost to the top, this would ensure they could take turns given the starting position of player E's piece falling before player A's.

As players play, in turn, through their initial configurations, only "bricks," the first block in Figure 1, fall. If they do not place them into the spaces that correspond to the choice of variables, they will immediately lose the game. Their only strategy is to place the bricks in the spaces labeled true and then false or false and then true before clearing the lines. The value corresponding to the space a player fills second is the player's setting of that variable. When player E clears two rows, he contributes to the filler on player A's board, shown in Figure 5, which appears two lines per two rows cleared. When player A clears two rows, also by setting a variable, two lines appear of the widget on player E's board. These lines are determined by how player E and player A each cleared their rows last. The columns in the widget are predetermined, except for the columns corresponding to the variables last set in the last rows cleared, where the path along the opposite of the value set becomes blocked off. This is illustrated by the darker blocks.

Finally, after the widget is fully formed, and after the variables have all been set, player A must choose the clause in the corresponding boolean formula. This leaves spaces on E's widget for the settings of the variables in the clause A chose. Finally, player E will get a "line," the second piece in Figure 1. If there is a path of empty spaces from the top of E's board to the bottom, then player E will win by clearing the bottom row with the line and creating enough rows on player A's board to win. If there is no such path, then the next time player A clears a row, this would make player E lose. For this reason, player A is allowed to choose the clause in the end. Since in a CNF formula, all clauses must be satisfied for the formula to be true, player A may choose any clause in the formula to examine, and if that clause is not true, it will block the empty space path on player E's board and win player A the game. Hence, player E has a winning strategy in this game if and only if he has a winning strategy in the corresponding FORMULA-GAME. We also note that the size of the m by n board is polynomial in the size of the FORMULA-GAME. This completes the reduction and shows 2PT is PSPACE hard. Since 2PT is PSPACE hard even if the players know the sequences of falling blocks in advance and even if they know the penalty blocks that will appear as a function of the rows they clear, it must be at least as hard if they are not given that information, as in real tetris. Whether 2PT is in PSPACE is yet unknown.

References

[1] Breukelaar, R., Demaine, E. D., Hohenberger, S., Hoogeboom, H. J., Kosters, W. A., and Liben-Nowell, D. Tetris is hard, even to approximate. *International Journal of Computational Geometry and Applications 14*, 1–2 (2004), 41–68.

[2] Lichtenstein, D., and Sipser, M. Go is pspace hard. In *FOCS* (1978), pp. 48–54.

[3] Sipser, M. *Introduction to the Theory of Computation.* International Thomson Publishing, 1996.