GRP_30: All Clear Solver

Greg Wang (19GBW1)

Steven Zhang (19SZ99)

Johnathan Wilson (19JMW19)

Brian Grigore (19BG1)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

October 30, 2021

Abstract

Tetris is likely the most recognizable video game in history, known for its very simplistic, easy to understand premise, though digging deeper there is a lot more to this game than first meets the eye. This application will solve in tetris what is known as an all clear, which is when, after the first move, the pieces on the board are all clear leaving an empty board behind. For the sake of terminology, when we discuss pieces later in this assignment we will use the word "tetrominio" as that is the official term.

Propositions

$p_i j$: This is true when tile (i,j) (Counting the bottom left as 1,1) is filled by a unit of a tetromino E.g. p1 1 - p1 6 would be filled in the example on the right
c_i: This is true when row i is filled with pieces and is cleared t_i: This is true when the piece currently being placed is type i E.g. I pieces are 1, L is 2, J is 3, O is 4, Z is 5, S is 6 and T is 7
a_i j: Is true where i is the position of the next piece in the queue and j is that tetromino's type E.g. On the right the O piece would be a4 4
r_i: Is true when the rotation of piece is r (1 is 90 degrees clockwise etc.) k_i j: Is true when piece's position while falling is (i, j)

Constraints

We want to model an all clear (or at least as close to one as possible) with the pieces we're given, this means: ¬p1 1 ∧ ¬p1 2∧ ¬p1 3∧ ¬p1 3 . . . ¬p20 10
Once a row is cleared its pieces are gone: c_i → (¬pi 1 ∧ ¬pi 2∧ . . . ¬pi 10) A row cleared proposition holds only when all spaces in it are filled

first: (pi 1 ∧ pi 2∧ . . . pi 10) → c_i

The current piece's rotation, type and position will determine what spaces are filled: E.g. (t7 ∧ r2 ∧ k20 7) → (p1 7 ∧ p2 6 ∧ p2 7 ∧ p2 8) on the right

Model Exploration

List all the ways that you have explored your model – not only the final version, but intermediate versions as well. See (C3) in the project description for ideas.

Jape Proof Ideas

$x \to (\neg z1 \land \neg z2 \land \neg z3), y1 \to z1, y2 \to z2, y3 \to z3, x \vdash \neg(y1 \land y2 \land y3)$:
Proving that if all a pieces tiles fall into a line being cleared, the piece does not exist anymore
$(P(x) \to (C(x) \land \exists y.(R(y) \to \neg R(y)))), P(x), \forall x.R(x) \vdash \exists x.\neg R(x)$:
Once a line is cleared, there will always be a row of tiles that gets shifted down because of the clear

Third proof: A boundary proof, if x is the current x position and y is a shift to the right a certain number of right moves leave the bounds

Fourth proof: Pieces that extend down will eventually have to stop falling at the bottom

Fifth proof: An all clear at once is the same effect as clearing all 20 rows individually

First-Order Extension

Describe how you might extend your model to a predicate logic setting, including how both the propositions and constraints would be updated. There is no need to implement this extension!

Useful Notation

Feel free to copy/paste the symbols here and remove this section before submitting.

$$\land \quad \lor \quad \neg \quad \to \quad \forall \quad \exists$$