Queen's UNIVERSITY

GRP_26: BattleShip

Frankie Cao 20341005

Christine Ma 20340949

Jacob Roe 20351389

Nicholas Wang 20364023

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

November 3, 2023

Abstract

This project is designed to evaluate the best possible guess at any given stage of a game of Battleship. Our model will interpret the known hits and misses from previous turns, and apply our understanding of probabilities to suggest the optimal next move.

Propositions

Maps:

Each player has two maps in battleship, those being the Primary, and Tracking maps, Both of which will be formatted as grids, with the x-axis being labeled A-J, and the y-axis being labeled 1-10.
The **primary** map serves to record hits / misses fired against the current player, and will be denoted as:

$$P_{PlayerNumber}(X - Coord, Y - Coord).$$

The **tracking** map serves to record hits / misses fired against the apposing player, and will be denoted as:

$$T_{PlayerNumber}(X - Coord, Y - Coord)$$

Guessing:

Guesses are taken on a turn based system, with each player guessing a coordinate to attack, once a player has guessed a coordinate, the apposing player will then respond with whether the guess was a hit or miss, evaluating to TRUE or FALSE respectively.
Guesses will be denoted as:

$$G(GuessNumber)_{PlayerNumber}(X - Coord, Y - Coord)$$

And will affect the current players Tracking Map, and the apposing players Primary Map based on the result of the guess.

Ships:

Each player has five ships that they may place before the guessing phase of the game takes place, these ships along with their respective lengths and letter notation are: **Carrier 5 (C), Battleship 4 (B), Destroyer 3 (D), Submarine 3 (S), and Patrol Boat 2 (P).**
Ships will be denoted as:

$$S_{PlayerNumber,ShipLetter}(X - Coord, Y - Coord)$$

Constraints

List of constraint types used in the model and their (English) interpretation.

You only need to provide one example for each constraint type: e.g., if you have constraints saying "cars have one colour assigned" in a car configuration setting, then you only need to show the constraints for a single car. Essentially,

we want to see the pattern for all of the types of constraints, and not every constraint enumerated.

Placement Constraints:

Ships must be placed so that all component coordinates of the ship are on the predefined map: ...

Ships may not be placed touching one another:

$$S_{P,L} \rightarrow \neg(S_{X,Y} \wedge ... \wedge S_{A,B})$$

Gameplay Constraints:

The game ends once all ships of either of the two players are sunk: ...
This constraint will be denoted as:

$$(S_{1C} \wedge S_{1B} \wedge S_{1D} \wedge S_{1S} \wedge S_{1P}) \vee (S_{2C} \wedge S_{2B} \wedge S_{2D} \wedge S_{2S} \wedge S_{2P})$$

A ship is destroyed once all of its component coordinates are hit where (a,b) is the starting coords of the ship and, (x,y) are the ending coords of the ship: ...
This constraint will be denoted as:

$$S_{P,L(a,b)} \wedge ... \wedge S_{P,L(x,y)} \rightarrow S_{P,L}$$

When a ship is destroyed, the coordinates around that ship cannot contain a ship and are "revealed" to the attacking player. An example for a horizontally orientated ship:

$$S_{P,L} \rightarrow \neg(S_{P,L(a-1,b+1)} \wedge ... \wedge S_{P,L(x+1,y+1)}) \wedge \neg(S_{P,L(a-1,b-1)} \wedge ... \wedge S_{P,L(x+1,y-1)}) \wedge \neg(S_{P,L(a-1,b)}) \wedge \neg(S_{P,L(x+1,y)})$$

Guessing Constraints:

A player may guess any coordinate defined within the confines of the map:

$$(G_P(A,1) \vee G_P(A,2) \vee \ldots \vee G_P(A,10)) \vee (G_P(B,1) \vee G_P(B,2) \vee \ldots \vee G_P(B,10)) \vee \ldots \vee (G_P(J,1) \vee G_P(J,2) \vee \ldots \vee G_P(J,10))$$

If a new ship is hit for the first time, the next guess should be in a cardinal direction relative to that guess:

$$G(n)_P(x,y) \wedge \neg S_L \rightarrow G(n+1)_P(x+1,y) \vee G(n+1)_P(x-1,y) \vee G(n+1)_P(x,y+1) \vee G(n+1)_P(x,y-1))$$

Once a ship is hit twice in some given direction, it can be assumes the ship is orientated in that direction, and the following guesses should be be consistent with that logic:

$$G(n-1)_P(x,y) \wedge G(n)_P(x+1,y) \wedge \neg S_L \rightarrow G(n+1)_P(x+2,y) \vee G(n+1)_P(x-1,y)$$

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.

First-Order Extension

Updated Propositions (Predicates):

P(x, y, z) : x and y are the coordinates of hits / misses against current player, z indicates player

T(x, y, z) : x and y are the coordinates of hits / misses against apposing player, z indicates player

G(x, y , z) : x and y are the coordinates guessed, z indicates player

S(w, x, y , z) : w is the type of ship, x and y are the coordinates ship touches, z indicates player

Updated Constraints:

$\forall w, x, y.S(w, x, y, z) \vee \forall a, b, c.S(a, b, c, d)$: End game requirement

$\forall x, y.S(w, x, y, z)$: Coordinates needed to hit to destroy ship

$\exists x, y.G(x, y, z)$: Guesses of coordinates defined by map