GROUP 3: Game Of Life

Emily Burke (22EB10)

Liam Ault (20LOJN)

Riche Cai (22TWQ4)

Martin Allerdissen (22SMJ3)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

October 31, 2024

Abstract

The game of life is set on a grid of alive and dead tiles. The state of the tiles can change from alive to dead or vice versa for every iteration of the game, based on the amount of neighboring alive tiles.

Our model will correspond to a random configuration on an 8x8 board and determine if the configuration will enter an infinite loop, become stable, or result in no tiles alive.

Propositions

1. T_{xyz}: represents one tile on an 8 x 8 grid, with coordinates (x,y). The z value is defined by the iteration of the border beginning at $z = 0$ and ending at $z = 9$. Each tile can take on the value of True or False, alive or dead, respectively, based on the following rules:

    - Remains true when T_{xyz} is neighboring 2 or 3 other true tiles.
    - Becomes false when T_{xyz} is surrounded by more than 3 or less than 2 true(alive) neighbors.
    - Becomes true when T_{xyz} is neighboring 3 other true tiles.

    Each tile T_{xyz} has a certain number of true (alive) and false (dead) neighbors, which influences its next state according to the rules of Conway's Game of Life.

2. S: represents stable, and is true when the current grid matches the subsequent grid, within 10 iterations.

3. L: represent a loop, and is true when for the current grid state, the same state repeats, without being stable within 10 iterations.

4. D: represents dead and is true when a configuration has no alive (true) tiles

Constraints

P: let P be the matrix $T_{000}, T_{100}, T_{110} \ldots T_{770}$ representing the initial configuration, then

D: Let D be True if all the tiles at some point between iteration 0 and 9 become all dead

$$((\neg T_{000} \wedge \ldots \wedge \neg T_{770}) \vee \ldots \vee (\neg T_{009} \wedge \ldots \wedge \neg T_{779})) \vdash D$$

S: Let S be True the state of all tiles match when comparing neighboring iterations

$$(T_{000} \leftrightarrow T_{001} \wedge \ldots \wedge T_{770} \leftrightarrow T_{771}) \vee \ldots \vee (T_{008} \leftrightarrow T_{009} \wedge \ldots \wedge T_{778} \leftrightarrow T_{779}) \vdash S$$

L: Let L be True if there are any matching configurations of T_{xyz} within z ¡ 10

$$((T_{000} \leftrightarrow T_{001} \wedge \ldots \wedge T_{770} \leftrightarrow T_{771}) \vee \ldots \vee (T_{008} \leftrightarrow T_{009} \wedge \ldots \wedge T_{778} \leftrightarrow T_{779}))\vee$$

$$((T_{001} \leftrightarrow T_{002} \wedge \ldots \wedge T_{771} \leftrightarrow T_{772}) \vee \ldots \vee (T_{008} \leftrightarrow T_{009} \wedge \ldots \wedge T_{778} \leftrightarrow T_{779})) \vee \ldots$$
$$\vee (T_{008} \leftrightarrow T_{009} \wedge \ldots \wedge T_{778} \leftrightarrow T_{779}) \vdash Lx$$

$$S \rightarrow L$$

$$\neg(L \rightarrow S)$$

Model Exploration

We chose to explore our model by exploring how an initial configuration changes based on the rules of a game, and what that the configuration represents after 10 iterations.

We want to determine if a configuration returns stable after 10 iterations, and explore this in propositional logic by modeling the implications and relationships between the propositions.

We are also curious about whether or not a configuration results in a loop, in other words, if a state repeats, but not consecutively. This is explored by first determining if the initial configuration is stable, which implies that there is a looping possibility.

Exploring loops further, we want to determine the first instance of a repeating frame, i.e. where the loop begins. We do this by modeling the constraints that result in a loop, and from this, we can derive the initial instance of the loop.

Moreover, we can explore the duration of a loop, for a constrained amount of iterations, in our case 10, and see how long the loop lasts. First we explore if our configuration ends up in a loop, and from this we can derive the quantity of iterations the loop occurs.

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!

- G_z : represents the grid configuration for all tiles T_{xy} on a grid, $z = 0$ is the initial configuration. G_z is the current configuration