OPTIMAL GARDENING WIKI

This is a complete and thorough walkthrough of the gardening system the Group 2 Logic Project aims to represent. Additionally, there is a complete list of all used constraints and propositions.

CONTENTS

GARDEN WALKTHROUGH

- The garden takes place on a square grid of arbitrary size. Each cell within the grid contains either a rock or a plant. Plants will, at any given time, be either alive or dead. All cells within the grid must contain a plant of some type. No cells may be empty. *Note: this means a partially empty garden will possess a certain amount of 'solutions' that are all possible configurations of plants in the empty cells*.
- The garden will change over time based on its configuration of plants. It will contain one unique grid for each time interval out of a certain number of time intervals (set by the gardener). The 'initial configuration' of a garden will be a configuration t_0 in which all plants are alive. All proceeding time intervals ($t_1, t_2, t_3, \ldots t_n$) will recursively relate to the initial configuration.
- All cells containing plants may at any time be marked as 'dead' or 'alive' depending on a variety of factors. Cells containing dead plants may become overridden by a living plant of an adjacent type. Dead plants will be overrun by the nearby plants they have the best relationship with, regardless of direction. Plant relationships and directionality are detailed later.
- The 'solution' to a particular garden configuration is the state of the garden in each time interval proceeding t_0. The only 'arbitrary' elements of the garden are, therefore, the features of the initial configuration of the garden. It should be noted that the *entire* state of a garden at t_n may be determined entirely by the state of t_{n-1}. That is, in general, every interval "implies" the next one.
- An 'optimal garden' is a garden that, at some time interval past t_0, will reach an 'optimal state', that is, all plants will remain alive and unmoved indefinitely. For example, if a garden contains various dead plants prior to t_6, but then contains the same configuration of living plants in all $t_7, t_8, t_9, \ldots$ and onward, it will be an 'optimal garden'. *Note: such a garden would only be 'optimal' for a duration of 8 or more. That is, a garden is optimal if its last two intervals possess the exact same garden configuration. If this is the case, then ALL proceeding intervals will also be exact.*
- A 'immediate optimal garden' is an even-more 'optimal' garden that *never* sees *any* plants die or move at any point in time. For this to be the case, t_0 must be identical to t_1. 'Immediate optimal gardens' never change over any interval, and all plants remain alive.

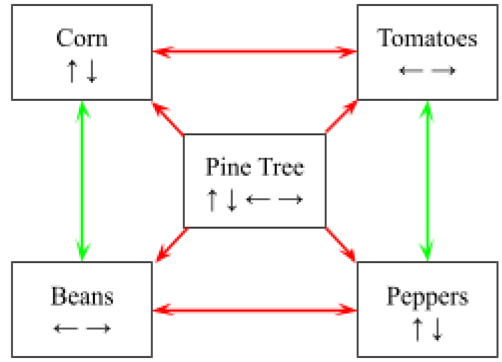
HOW TO BUILD A GARDEN

- There are five objects one may place in the garden: Rocks, Pine trees, Tomatoes, Beans, Corn, and Peppers. The exact plant relationships are listed on the chart below.
- Rocks are used as 'boundary' objects. They cannot be helped or harmed. Rocks will *technically* be hard-coded as dead, but this will have *no* effect on whether a garden is optimal in any way.
- If a plant is both helped and hurt, it will remain alive. If it is only hurt, it will die. That is, all plants obtain the most positive possible state at each time interval. This means Pine Trees will never die, since nothing can hurt them. This is an important factor in setup, as the optimization of your garden will be drastically influenced by the presence of any amount of Pine Trees.
- Each plant has a certain 'direction' in which it may help/hurt other plants. The exact directions are indicated below. For example, a Corn plant will kill Tomatoes in cells to its left/right, but not those above/below. Pine trees hurt plants in all four cell directions.

PLANT RELATIONSHIPS

Plant type	Helps	Hurts	Direction
Corn	Beans	Tomatoes	Vertical
Beans	Corn	Peppers	Horizontal
Tomatoes	Peppers	Corn	Horizontal
Peppers	Tomatoes	Beans	Vertical
Pine Trees	-	Beans, Corn, Tomatoes, Peppers	Horizontal, Vertical
Rock	-	-	-

Alternative Visualization:



PLANT SOLUTIONS/SPREADING

- The gardener will only voluntarily set the initial placements of the plants in t_0. All following intervals will be derived from this initial state.
- Plants will be hurt/helped/killed based on plant placement of the *previous* interval. That is, the entire state of a garden at interval t_{n+1} may be determined solely by the state of the interval t_n. The only exception to this is t_0 in which all plants are alive.
- If a cell contains a *dead* plant at a given interval, then that cell may be overridden with a new plant during the *next* interval. The plant type which will override the dead plant will be the adjacent type (in any of the 4 directions) that obtains the *best* relationship with the dead plant type. For example, if a dead Corn is next to Beans and Tomatoes, the cell will be overridden with a Bean plant in the next interval. Plants may NOT override themselves. Pine trees may NOT spread. A dead plant may remain dead if no available plants override it on a given interval.
- In general, the plant override selection goes as follows with regard to priority:
    - Helpful plant > Indifferent plant > Hurtful plant

GARDEN FEATURES TO EXPLORE

- The following features or calculations may be explored as part of a logical exploration of the optimal garden system detailed above:
    - How many optimal gardens are there given a certain configuration of pine trees?
    - Given that a garden is optimal, what must the plant at cell (x,y) be?
    - What amount of 'optimal gardens' are 'immediate'?

LOGICAL REPRESENTATION OF A GARDEN

PROPOSITIONS All propositions are organized into “cells” in the square garden grid. Each one of these cells has each one of the following propositions.	
$PT_{(x,y,t)}$	Whether the cell at (x,y) has a pine tree at time interval t.
$C_{(x,y,t)}$, $B_{(x,y,t)}$, $T_{(x,y,t)}$, $P_{(x,y,t)}$	Whether the cell at (x,y) has a plant of the given type at time interval t: C - Corn, B - Beans, T - Tomatoes, P - Peppers
$R_{(x,y,t)}$	If the cell at (x,y) is a rock at time interval t
$h_{(x,y,t)}$	If the plant at (x,y) is being helped by adjacents at time interval t
$k_{(x,y,t)}$	If the plant at (x,y) is being hurt by adjacents at time interval t
$a_{(x,y,t)}$	If the plant at (x,y) is alive at time interval t

SEQUENTIAL CONSTRAINTS These constraints have antecedents at some interval t_n and consequents at t_{n+1}.	
$C_{(x,y\pm1,t)} \land B_{(x,y,t)} \rightarrow h_{(x,y,t+1)}$ $B_{(x\pm1,y,t)} \land P_{(x,y,t)} \rightarrow k_{(x,y,t+1)}$ … etc (see relationships chart)	Plants help or hurt certain plant types based on their respective direction. This effect will be carried out in the next time interval.
$C_{(x,y,t)} \land \neg a_{(x,y,t)} \land (B/T/P)_{(x\pm1,y,t)}$ $\rightarrow (B/T/P)_{(x,y,t+1)}$ $C_{(x,y,t)} \land \neg a_{(x,y,t)} \land \neg(B/T/P)_{(x\pm1,y,t)}$ $\rightarrow C_{(x,y,t+1)} \land \neg a_{(x,y,t)}$ … etc (see plant spreading)	On the next time interval, a dead plant will be overridden by the highest-priority plant adjacent to it in any direction (if such a plant exists). If no plant can override the dead plant, it will stay dead. Dead plants cannot be revived unless overridden.
$R_{(x,y,t)} \rightarrow R_{(x,y,t+1)}$ $PT_{(x,y,t)} \rightarrow PT_{(x,y,t+1)}$	Rocks and Pine Trees stay the same throughout all time intervals. That is, if there is one at (x,y), it will always be at (x,y).
$C_{(x,y,t)} \land a_{(x,y,t)} \rightarrow C_{(x,y,t+1)}$ … etc	Normal plants (Corn/Tomato/Beans/Peppers) will not be overrun if they are not dead (that is, they stay the same).

SIMPLE CONSTRAINTS These constraints are common to all time intervals, and all cells in them. They have no effect on cells surrounding a particular cell, and are focused on one particular point.	
$PT \lor C \lor B \lor T \lor P \lor R$ $PT \rightarrow \neg C \land \neg B \land \neg T \land \neg P \land \neg R$ … and so on	A cell must have one, and only one, plant at any given time.
$(h \lor \neg k) \land \neg R \rightarrow a$ $a \rightarrow (h \lor \neg k) \land \neg R$	A plant will be alive *if and only if* it isn’t a Rock, and it is helped or not hurt. More simply, non-Rock plants die if they are *only harmed*.
$PT \rightarrow \neg h \land \neg k \land a$	Pine Trees are never harmed or helped, and are always alive.
$R \rightarrow \neg h \land \neg k \land \neg a$	Rocks are never harmed or helped, and are never alive.

LOGICAL REPRESENTATION OF A GARDEN (cont.)

OPTIMAL CONSTRAINTS These constraints force a garden to be optimal. The user may turn them on or off depending on what gardens they wish to discover.	
And(($a_{(x,y,n-1)} \wedge a_{(x,y,n)}$) $\vee$ $R_{(x,y,t)}$ … for all cells (x,y) at last two time intervals, *n* and *n-1*	*For basic optimization*: The final two time intervals will *both* include entirely living plants. (If this holds, then we *know* no plants are left to kill or be overrun, meaning it will *stay* this way). Rocks are not considered in optimization, as they are always "not alive".
And(($a_{(x,y,0)} \wedge a_{(x,y,1)}$) $\vee$ $R_{(x,y,0)}$ … for all cells (x,y) at first two time intervals, t_0 and t_1	*For immediate optimization*: The first two time intervals will *both* include entirely living plants. Same reasoning as above with regard to plant permanence.

ADDITIONAL LINKS

GitHub repo: https://github.com/sirivanbiscuit/optimal-gardening