Reduction Visualisation

A-O-S - 27.08.24

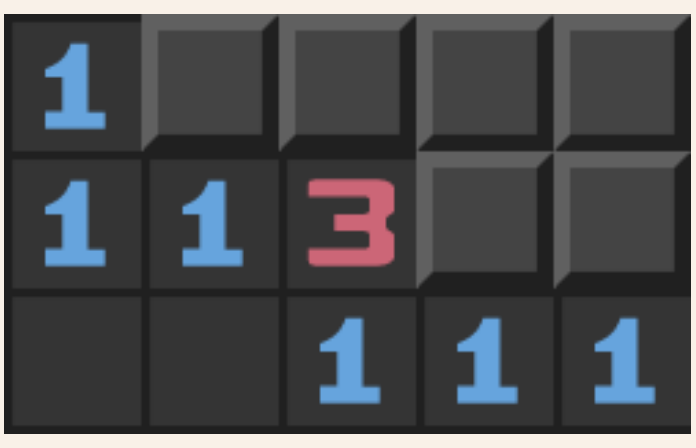
General

Reduction in Minesweeper is used to simplify the logic of situations.
Specifically, this is done by temporarily ignoring cells that aren't relevant to current logic, considering groups of cells we know contain a certain number of mines
These groups will be referred to as "sets".

Reduction is never a *requirement* when solving, you can always just keep track of every mine relevant to the logic you are working on, so it's possible to become very skilled without giving this a closer look.

Though, it is worth studying, especially for practice with higher density boards and more complicated logic.
I strongly urge you to have a look over these examples, then employ this way of thinking in your own games, see if it changes how you visualise your gameplay.

In this example, it's fairly trivial to complete without any thoughtful reduction, but it illustrates the basic idea.



The logic we will apply reduction to.

Identify any known mines, or acknowledge any that were found previously.

Finally:
"subtract" or "**reduce**" all the mines from the surrounding clear cells showing numbers.
Notice how this effects both the **3** and **1** cells, reducing them into **2** and **0** respectively.

The end goal is to be able to visualise this.
Truthfully try to **see** the **3** as a **2** or more usefully, that the yellow cells contain 2 mines between them.
You want this process to become second nature.

As stated, this situation is trivial, but I encourage you to study this, or even play a low density board now and try to see basic situations like this, even if you naturally wouldn't look for them.

Practical Uses

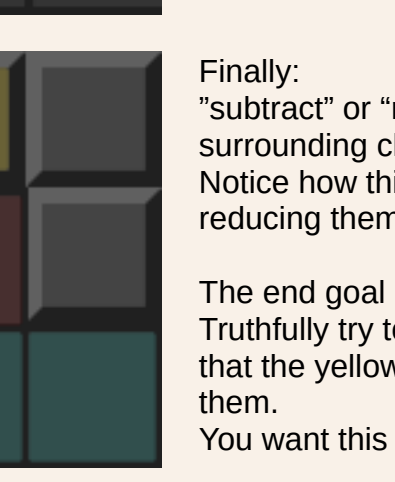
I believe there are 2 types of logic in which it is profitable to use reduction:
-Memorized patterns – *See the reduction section **here**, this is already explained so I will not cover it.*
-Reducing with a **set** of cells that contains a number of mines, not just individual mines.

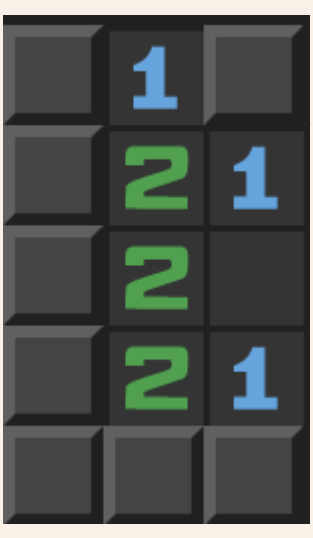
By this, I mean: If you can **guarantee** a cell is a mine (*if it can be correctly flagged*) then the effort of performing reduction outweighs the cost of simply keeping track of the mines.
Look at the previous example, understand why it is inefficient to reduce there.

Unknown set example

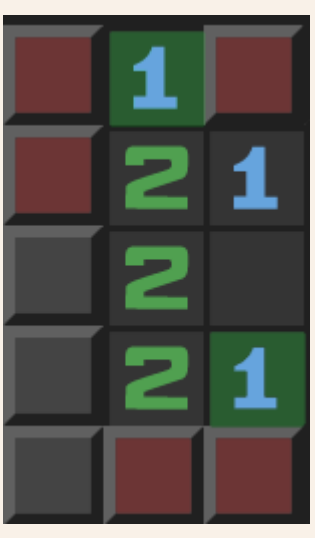
The real beauty / benefit of reduction comes from when we **don't** know precisely what cells are mines.

To aid visualisation, here is the **entire board** before we zoom in on the logic.
There are more simple steps possible, for example the bottom right **3**, but we are focusing on reduction here.





Initial logic



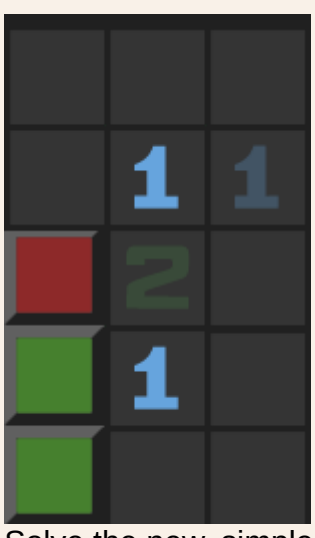
Define set of cells containing mines.
In this case each has exactly 1 mine in.

Perform the reduction.
Look how much easier this is!
If you learn to do this in your head, you improve greatly.

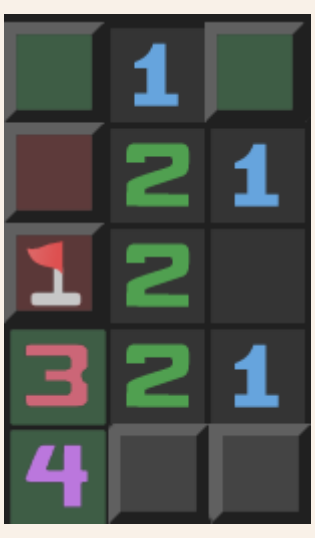
Perform the reduction*
Any **number** cell that is adjacent to all the cells in a set: you may "**reduce**" by subtracting the mine count of that set from the cell.

all cells in a set will be **unopened**.

A very important note: Notice how the faded out **1** and **2** have **not** been reduced. This is because not every cell in the set (the top 3 red cells) are adjacent to those cells, so there is no guarantee we can abstract the mines away from these. It is also important to "fade" or "ignore" these mentally, since they do not reflect the number of mines in the uncovered cells remaining (after reducing).
We only want to focus on the numbers we **have** reduced.



Solve the new, simple logic



Apply our derivations to the original board.
From which we can proceed as usual.

This example uses 2 sets for illustration, using the top set alone is sufficient to find a mine, but the 2 safe cells can be found neatly with another one.

Most practical cases will require only 1 set.

Board with all (currently found) certain mines reduced example

1-2-1 pattern found and used, solution is then applied to the board.