Complexity of Minesweeper with Restricted Number Sets

by

Ray Hua Wu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

Author ....
Department of Electrical Engineering and Computer Science
2018 February 09

Certified by ....
Prof. Erik D. Demaine
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by ....
Dr. Christopher Terman
Chairman, Masters of Engineering Thesis Committee

Complexity of Minesweeper with Restricted Number Sets

by

Ray Hua Wu

Submitted to the Department of Electrical Engineering and Computer Science
on 2018 February 09, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

We consider the Minesweeper consistency problem (is a given partially completed board consistent with some mine placement?) when the set of numbers that may appear on a Minesweeper board is restricted. First, we analyze the possible sets of numbers that could exist in legal rectangular Minesweeper boards, proving either possibility or impossibility for 509 of the 512 subsets of $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ (leaving 3 subsets as open problems), and thus make conclusions on the relations among some restricted-set Minesweeper consistency problems. We prove either inclusion in P or NP-completeness for the restricted-set Minesweeper consistency problem for 134 of the 512 subsets of the set of numbers above. In particular, we show that {0,1}-Minesweeper consistency is NP-complete, while {0}-Minesweeper consistency and {1}-Minesweeper consistency are in P. We also suggest a few more dimensions in which the Minesweeper consistency problem could be analyzed.

Thesis Supervisor: Prof. Erik D. Demaine
Title: Professor of Computer Science and Engineering

Acknowledgments

I would like to thank my advisor, Erik D. Demaine, for the generous guidance and relevant knowledge he provided for my thesis. I would also like to thank the EECS Department of MIT for their facilitation, particularly Anne Hunter for her wonderful and effective help with logistical issues. In addition, I would like to thank peers that helped with LaTeX issues, particularly Anders Kaseorg and Alex Chernyakhovsky, who provided significant assistance.

Contents

1 Introduction **13**

2 Validity of Ranges **17**

2.1 Summary of Results 17

2.2 Mine-Bordered Boards 18

2.2.1 Base Cases 18

2.2.2 Union Theorem 23

2.3 Zero-Bordered Boards 24

2.3.1 Base Cases 24

2.3.2 Union Theorem 25

2.4 Wall Boards 26

2.4.1 Union Theorem 26

2.4.2 Construction of Wall Cases 27

2.5 Individual Cases 28

2.6 Gap Impossibility Theorems 29

3 Complexity of the Consistency Problem in Modified Minesweeper Games **37**

3.1 Easy Cases: {}-Minesweeper, {0}-Minesweeper, and {8}-Minesweeper 37

3.2 {1}-Minesweeper 39

3.3 {0,1}-Minesweeper 42

3.4 Implied Results 44

3.5 Relating Problems Using Ranges 44

4 Further Questions **47**

4.1 Still-Open Restricted-Range Minesweeper Problems 47

4.2 Post- versus Pre- Restricted Set Minesweeper Consistency 48

4.3 Other Minesweeper Problems 48

4.4 Infinite Boards 49

4.5 Non-Rectangular Boards 50

4.6 Boards in More than Two Dimensions or Other Topologies 50

List of Figures

1-1 A completed 5x5 Minesweeper board with two 0s, one internal and one in a corner. 14

2-1 Base cases of mine-bordered boards. 23
2-2 Base cases of zero-bordered boards. 25
2-3 Boards for individually-proven ranges. 29
2-4 Three cells closer to B than A is when A and B are horizontally aligned. 30
2-5 Three cells closer to B than A is when A and B are not horizontally or vertically aligned. 31

3-1 wire 43
3-2 turn gadget 43
3-3 split gadget 43
3-4 phase shift gadget 43
3-5 clause gadget 43

List of Tables

2.1 Which of the 512 ranges are impossible (imp), unsolved (open), or possible (any other label). The cells also detail how the result is proved, as detailed in Section 2.1. 19

2.2 Corresponding MBBs and ZBBs for ranges proven possible with the Wall Union Theorem (Section 2.4.1). 27

3.1 Complexity of S-Minesweeper consistency for the 512 possibilities of S. 38

Chapter 1

Introduction

Minesweeper is a computer puzzle game originating from the 1960s and popularized by its inclusion in Microsoft Windows [10]. It involves a grid of cells, an unknown subset of which contain mines. The player interacts with the game by clicking cells. Clicking a cell with a mine loses the game. Clicking a cell without a mine reveals a number which indicates the total number of the adjacent cells containing mines. The game is won when all squares are successfully identified as bearing a mine or not. The Windows versions of Minesweeper interpret this as the player clicking every square that isn't a mine [11].

For our purposes, we will assume a Minesweeper board to be rectangular, as is standard. This implies, among other things, that all numbers in the completed board are between 0 and 8 inclusive. (In typical computer implementations of Minesweeper, 0s are shown not as 0s, but simply as depressed squares—see Figure 1. In fact, clicking a 0 is often processed to include revealing all surrounding squares, recursively if there are also 0s in adjacent squares, because such moves are guaranteed to be safe.)

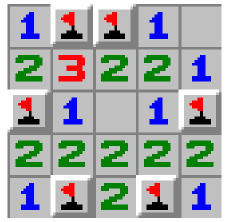


Figure 1-1: A completed 5x5 Minesweeper board with two 0s, one internal and one in a corner.

Minesweeper consistency [3] is the following decision problem: given an assignment of numbers and mines to some subset of a Minesweeper board, determine whether there exists some completed board that is consistent with the entities revealed. That is, does there exist a complete Minesweeper board with specifically the numbers and mines given in the respective squares, where each number in fact truthfully specifies the number of mines in adjacent squares?

Kaye [3] proved that the Minesweeper consistency problem is NP-complete via reduction from Circuit-SAT, constructing gadgets to emulate wires and logic gates. Examining these gadgets, we notice that the reduction does not require all numbers (0 through 8). This motivates the question studied in this thesis: how far can we restrict the set of numbers allowed in our Minesweeper boards and have the Minesweeper consistency problem remain NP-complete?

More specifically, consider the $2^9 = 512$ subsets of the integers 0 through 8 inclusive. For which of these subsets of allowable numbers does Minesweeper consistency remain NP-complete, and for which of these is the problem in P? Let's denote S-Minesweeper as the variant of Minesweeper where only numbers in the set S can appear on the completed board. As Kaye's reduction involves no gadgets containing a number higher than 5, Kaye's reduction shows $\{0, 1, 2, 3, 4, 5\}$-Minesweeper consistency NP-complete. Could we make do with an even smaller subset S?

The daunting task of categorizing the complexity of 512 subsets is aided somewhat by a few properties. If S-Minesweeper consistency is in P, and $T \subset S$, then T-Minesweeper consistency is in P. Likewise, if S-Minesweeper consistency is NP-complete, then so is T-Minesweeper consistency for $T \supset S$. Thus Kaye's reduction allows us to categorize not only $\{0, 1, 2, 3, 4, 5\}$-Minesweeper consistency as NP-complete, but also the $2^3 - 1 = 7$ variants of Minesweeper consistency for proper supersets of $\{0, 1, 2, 3, 4, 5\}$.

Further, as we will show, some sets of numbers are impossible as an exact set of numbers on a valid Minesweeper board. We first determine, for each of the subsets of the integers 0 through 8, whether that subset is in fact possible or not (Section 2). Next, we will prove NP-completeness or membership in P for the Minesweeper variants with particular subsets (Section 3). For S-Minesweeper consistency where S is an impossible set of numbers, our problem becomes the question of whether any proper subset of S is the set for an NP-complete version of Minesweeper consistency: if any such subset exists, S-Minesweeper consistency is NP-complete; otherwise, it's in P. Note that one cannot just immediately conclude that S-Minesweeper consistency is in P when S is an impossible set of numbers: although that set is impossible, a subset of it could be possible and also have a corresponding NP-complete consistency problem.

Chapter 2

Validity of Ranges

Definition 1. *The range of a completed Minesweeper board B is the set of all numbers k for which at least one cell on B contains k.*

Definition 2. *A valid range is a set that is the range of at least one valid completed Minesweeper board.*

2.1 Summary of Results

In Table 2.1, for each subset of $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$:

- If the cell for the subset contains **imp**, it is impossible for a valid Minesweeper board to have that exact subset as the set of numbers that appears at least once on the board, as proved by the Gap Impossibility Theorem given in parentheses, and thus, the represented set of numbers is not a valid range. For example, the set $\{1, 2, 8\}$ has a corresponding cell in the chart that displays "imp(2-8)", indicating that the 2-8 Gap Impossibility Theorem proves $\{1, 2, 8\}$ an invalid range. These Gap Impossibility Theorems are proved in Section 2.6.

- All other cells except the three marked "open" correspond to valid ranges.

1. Cells marked **MBB** correspond to ranges proved valid by the Mine-Bordered Boards Union Theorem (Section 2.2), along with the parenthesized num-

bered base case boards that are unioned together to prove the set of numbers possible. In the case of just one number in parentheses, the set of numbers corresponds exactly to the set of numbers on that precise base-case mine-bordered board.

2. Cells marked **ZBB** correspond to ranges proved valid by the Zero-Bordered Boards Union Theorem (Section 2.3). The following parenthesized numbers follow the same convention as with the MBB cells, but using the Zero-Bordered Boards base cases.

3. Cells marked **WALL** correspond to ranges proved valid by the Wall Union Theorem (Section 2.4). For this theorem, the specific proofs for each case are provided in the section describing the theorem itself (specifically, in 2.4.2), to aid in preventing the chart from being too large to fit on a page.

4. Cells marked **IND** correspond to ranges proved valid individually, with the parenthesized number indicating which board in Individually Proven Cases (Section 2.5) corresponds to that range.

5. The three cells marked **open** correspond to the ranges that have been neither proved valid nor proved invalid.

2.2 Mine-Bordered Boards

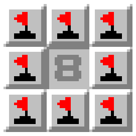
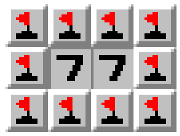
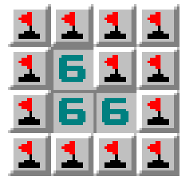
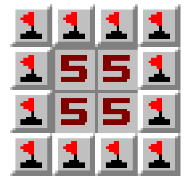
Definition 3. *A mine-bordered board is a board where every cell on each of its edges contains a mine, that is, all cells in the first row, last row, first column, and last column are mines.*

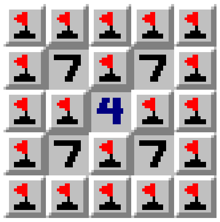
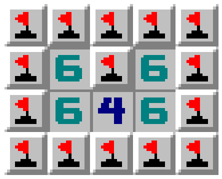
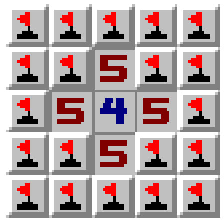
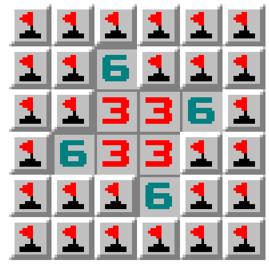
2.2.1 Base Cases

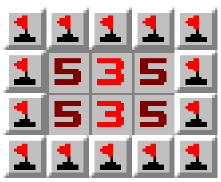
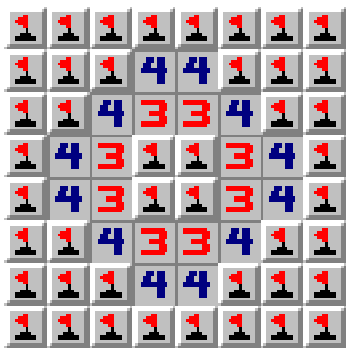
Figure 2-1 displays the 29 base cases of mine-bordered boards.

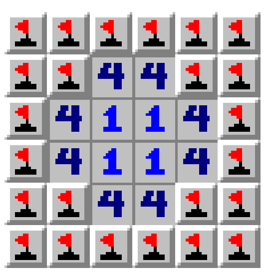
∪	{6,7,8}	{7,8}	{6,8}	{8}	{6,7}	{7}	{6}	{}
{0,1,2,3,4,5}	MBB(1,6,11,27)	MBB(1,7,11,27)	MBB(1,7,11,28)	MBB(1,10,13,21)	MBB(6,11,27)	MBB(7,11,27)	MBB(7,11,28)	MBB(10,13,21)
{1,2,3,4,5}	MBB(1,5,15,19)	MBB(1,2,10,19)	MBB(1,3,10,19)	MBB(1,10,19)	MBB(5,15,19)	MBB(2,10,19)	MBB(3,10,19)	MBB(10,19)
{0,2,3,4,5}	MBB(1,5,8,21)	MBB(1,2,12,20)	MBB(1,3,12,20)	MBB(1,12,20)	MBB(5,8,21)	MBB(2,12,20)	MBB(3,12,20)	MBB(12,20)
{2,3,4,5}	MBB(1,5,8,11)	MBB(1,5,9,11)	MBB(1,6,8,11)	MBB(1,7,9,11)	MBB(5,8,11)	MBB(5,9,11)	MBB(6,8,11)	MBB(7,9,11)
{0,1,3,4,5}	MBB(1,6,7,27)	MBB(1,2,13,20)	MBB(1,3,13,20)	MBB(1,13,20)	MBB(6,7,27)	MBB(2,13,20)	MBB(3,13,20)	MBB(13,20)
{1,3,4,5}	MBB(1,2,3,9,13)	MBB(1,2,9,13)	MBB(1,3,9,13)	MBB(1,9,13)	MBB(2,3,9,13)	MBB(2,9,13)	MBB(3,9,13)	MBB(9,13)
{0,3,4,5}	MBB(1,2,3,7,20)	MBB(1,2,7,20)	MBB(1,3,7,20)	MBB(1,7,20)	MBB(2,3,7,20)	MBB(2,7,20)	MBB(3,7,20)	MBB(7,20)
{3,4,5}	MBB(1,2,3,7,9)	MBB(1,2,7,9)	MBB(1,3,7,9)	MBB(1,7,9)	MBB(2,3,7,9)	MBB(2,7,9)	MBB(3,7,9)	MBB(7,9)
{0,1,2,4,5}	MBB(1,5,19,26)	MBB(1,2,13,21)	MBB(1,3,13,21)	MBB(1,13,21)	MBB(5,19,26)	MBB(2,13,21)	MBB(3,13,21)	MBB(13,21)
{1,2,4,5}	MBB(1,2,3,7,19)	MBB(1,2,7,19)	MBB(1,3,7,19)	MBB(1,7,19)	MBB(2,3,7,19)	MBB(2,7,19)	MBB(3,7,19)	MBB(7,19)
{0,2,4,5}	MBB(1,2,3,7,21)	MBB(1,2,7,21)	MBB(1,3,7,21)	MBB(1,7,21)	MBB(2,3,7,21)	MBB(2,7,21)	MBB(3,7,21)	MBB(7,21)
{2,4,5}	MBB(1,2,3,7,12)	MBB(1,2,7,12)	MBB(1,3,7,12)	MBB(1,7,12)	MBB(2,3,7,12)	MBB(2,7,12)	MBB(3,7,12)	MBB(7,12)
{0,1,4,5}	MBB(1,2,3,7,22)	MBB(1,2,7,22)	MBB(1,3,7,22)	MBB(1,7,22)	MBB(2,3,7,22)	MBB(2,7,22)	MBB(3,7,22)	MBB(7,22)
{1,4,5}	MBB(1,2,3,7,13)	MBB(1,2,7,13)	MBB(1,3,7,13)	MBB(1,7,13)	MBB(2,3,7,13)	MBB(2,7,13)	MBB(3,7,13)	MBB(7,13)
{0,4,5}	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)
{4,5}	MBB(1,2,3,7)	MBB(1,2,7)	MBB(1,3,7)	MBB(1,7)	MBB(2,3,7)	MBB(2,7)	MBB(3,7)	MBB(7)
{0,1,2,3,5}	MBB(1,3,11,27)	MBB(1,2,19,20)	MBB(1,3,19,20)	MBB(1,19,20)	MBB(3,11,27)	MBB(2,19,20)	MBB(3,19,20)	MBB(19,20)
{1,2,3,5}	MBB(1,2,8,19)	MBB(1,2,18,19)	MBB(1,3,18,19)	MBB(1,18,19)	MBB(2,8,19)	MBB(2,18,19)	MBB(3,18,19)	MBB(18,19)
{0,2,3,5}	MBB(1,2,3,9,21)	MBB(1,2,9,21)	MBB(1,3,9,21)	MBB(1,9,21)	MBB(2,3,9,21)	MBB(2,9,21)	MBB(3,9,21)	MBB(9,21)
{2,3,5}	MBB(1,2,3,9,11)	MBB(1,2,9,11)	MBB(1,3,9,11)	MBB(1,9,11)	MBB(2,3,9,11)	MBB(2,9,11)	MBB(3,9,11)	MBB(9,11)
{0,1,3,5}	MBB(1,3,4,27)	MBB(1,2,18,20)	MBB(1,3,18,20)	MBB(1,18,20)	MBB(3,4,27)	MBB(2,18,20)	MBB(3,18,20)	MBB(18,20)
{1,3,5}	MBB(1,2,3,18)	MBB(1,2,18)	MBB(1,3,18)	MBB(1,18)	MBB(2,3,18)	MBB(2,18)	MBB(3,18)	MBB(18)
{0,3,5}	MBB(1,2,3,20)	MBB(1,2,20)	MBB(1,3,20)	MBB(1,20)	MBB(2,3,20)	MBB(2,20)	MBB(3,20)	MBB(20)
{3,5}	MBB(1,2,3,9)	MBB(1,2,9)	MBB(1,3,9)	MBB(1,9)	MBB(2,3,9)	MBB(2,9)	MBB(3,9)	MBB(9)
{0,1,2,5}	MBB(1,2,3,19,21)	MBB(1,2,19,21)	MBB(1,3,19,21)	MBB(1,19,21)	MBB(2,3,19,21)	MBB(2,19,21)	MBB(3,19,21)	MBB(19,21)
{1,2,5}	MBB(1,2,3,19)	MBB(1,2,19)	MBB(1,3,19)	MBB(1,19)	MBB(2,3,19)	MBB(2,19)	MBB(3,19)	MBB(19)
{0,2,5}	MBB(1,2,3,21)	MBB(1,2,21)	MBB(1,3,21)	MBB(1,21)	MBB(2,3,21)	MBB(2,21)	MBB(3,21)	MBB(21)
{2,5}	MBB(1,2,3,11)	MBB(1,2,11)	MBB(1,3,11)	MBB(1,11)	MBB(2,3,11)	MBB(2,11)	MBB(3,11)	MBB(11)
{0,1,5}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)
{1,5}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)
{0,5}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)
{5}	MBB(1,2,3,4)	MBB(1,2,4)	MBB(1,3,4)	MBB(1,4)	MBB(2,3,4)	MBB(2,4)	MBB(3,4)	MBB(4)
{0,1,2,3,4}	MBB(1,5,11,28)	MBB(1,12,27)	MBB(1,12,28)	MBB(1,10,12,22)	MBB(5,11,28)	MBB(12,27)	MBB(12,28)	MBB(10,12,22)
{1,2,3,4}	MBB(1,6,12,16)	MBB(1,12,16)	MBB(1,12,17)	MBB(1,10,12,13)	MBB(6,12,16)	MBB(12,16)	MBB(12,17)	MBB(10,12,13)
{0,2,3,4}	MBB(1,23,26)	MBB(1,12,23)	MBB(1,12,24)	WALL	MBB(23,26)	MBB(12,23)	MBB(12,24)	WALL
{2,3,4}	MBB(1,2,3,10,12)	MBB(1,2,10,12)	MBB(1,3,10,12)	MBB(1,10,12)	MBB(2,3,10,12)	MBB(2,10,12)	MBB(3,10,12)	MBB(10,12)
{0,1,3,4}	MBB(1,2,3,10,22)	MBB(1,2,10,22)	MBB(1,3,10,22)	MBB(1,10,22)	MBB(2,3,10,22)	MBB(2,10,22)	MBB(3,10,22)	MBB(10,22)
{1,3,4}	MBB(1,2,3,10,13)	MBB(1,2,10,13)	MBB(1,3,10,13)	MBB(1,10,13)	MBB(2,3,10,13)	MBB(2,10,13)	MBB(3,10,13)	MBB(10,13)
{0,3,4}	MBB(1,2,24)	MBB(1,23)	MBB(1,24)	IND(17)	MBB(2,24)	MBB(23)	MBB(24)	IND(16)
{3,4}	MBB(1,2,3,10)	MBB(1,2,10)	MBB(1,3,10)	MBB(1,10)	MBB(2,3,10)	MBB(2,10)	MBB(3,10)	MBB(10)
{0,1,2,4}	MBB(1,2,3,12,22)	MBB(1,2,12,22)	MBB(1,3,12,22)	MBB(1,12,22)	MBB(2,3,12,22)	MBB(2,12,22)	MBB(3,12,22)	MBB(12,22)
{1,2,4}	MBB(1,2,3,12,13)	MBB(1,2,12,13)	MBB(1,3,12,13)	MBB(1,12,13)	MBB(2,3,12,13)	MBB(2,12,13)	MBB(3,12,13)	MBB(12,13)
{0,2,4}	MBB(1,2,26)	MBB(1,25)	MBB(1,26)	IND(19)	MBB(2,26)	MBB(25)	MBB(26)	IND(18)
{2,4}	MBB(1,2,3,12)	MBB(1,2,12)	MBB(1,3,12)	MBB(1,12)	MBB(2,3,12)	MBB(2,12)	MBB(3,12)	MBB(12)
{0,1,4}	MBB(1,2,3,22)	MBB(1,2,22)	MBB(1,3,22)	MBB(1,22)	MBB(2,3,22)	MBB(2,22)	MBB(3,22)	MBB(22)
{1,4}	MBB(1,2,3,13)	MBB(1,2,13)	MBB(1,3,13)	MBB(1,13)	MBB(2,3,13)	MBB(2,13)	MBB(3,13)	MBB(13)
{0,4}	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)	imp(0-4)
{4}	MBB(1,2,6)	MBB(1,5)	MBB(1,6)	IND(1)	MBB(2,6)	MBB(5)	MBB(6)	IND(2)
{0,1,2,3}	MBB(1,14,28)	MBB(1,14,27)	MBB(1,15,28)	ZBB(2,5)	MBB(14,28)	MBB(14,27)	MBB(15,28)	ZBB(2,3)
{1,2,3}	MBB(1,14,17)	MBB(1,14,16)	MBB(1,15,17)	IND(21)	MBB(14,17)	MBB(14,16)	MBB(15,17)	IND(20)
{0,2,3}	WALL	WALL	WALL	WALL	WALL	WALL	WALL	WALL
{2,3}	MBB(1,2,15)	MBB(1,14)	MBB(1,15)	WALL	MBB(2,15)	MBB(14)	MBB(15)	WALL
{0,1,3}	MBB(1,2,28)	MBB(1,27)	MBB(1,28)	ZBB(5)	MBB(2,28)	MBB(27)	MBB(28)	ZBB(3)
{1,3}	MBB(1,2,16)	MBB(2,16)	MBB(1,16)	IND(3)	MBB(16,17)	MBB(16)	MBB(17)	IND(4)
{0,3}	imp(0-3)	imp(0-3)	imp(0-3)	imp(0-3)	imp(0-3)	imp(0-3)	imp(0-3)	imp(0-3)
{3}	MBB(1,2,8)	IND(5)	MBB(1,8)	IND(7)	MBB(2,8)	IND(6)	MBB(8)	IND(8)
{0,1,2}	IND(26)	IND(25)	IND(22)	imp(2-8)	IND(24)	IND(23)	ZBB(4)	ZBB(2)
{1,2}	IND(33)	IND(32)	IND(31)	imp(2-8)	IND(30)	IND(29)	IND(28)	IND(27)
{0,2}	open	open	open	imp(2-8)	IND(35)	IND(36)	IND(34)	IND(15)
{2}	IND(9)	IND(10)	IND(13)	imp(2-8)	IND(11)	IND(12)	IND(14)	IND(37)
{0,1}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	ZBB(1)
{1}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	IND(38)
{0}	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	imp(1-5)	ZBB(0)
{}	MBB(1,2,3)	MBB(1,2)	MBB(1,3)	MBB(1)	MBB(2,3)	MBB(2)	MBB(3)	MBB(0)

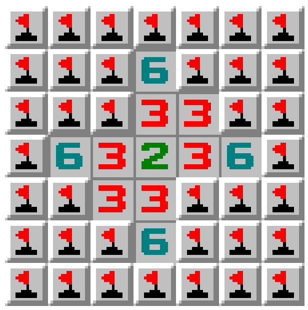
Table 2.1: Which of the 512 ranges are impossible (imp), unsolved (open), or possible (any other label). The cells also detail how the result is proved, as detailed in Section 2.1.

0.  1.  2.  3.  4. 

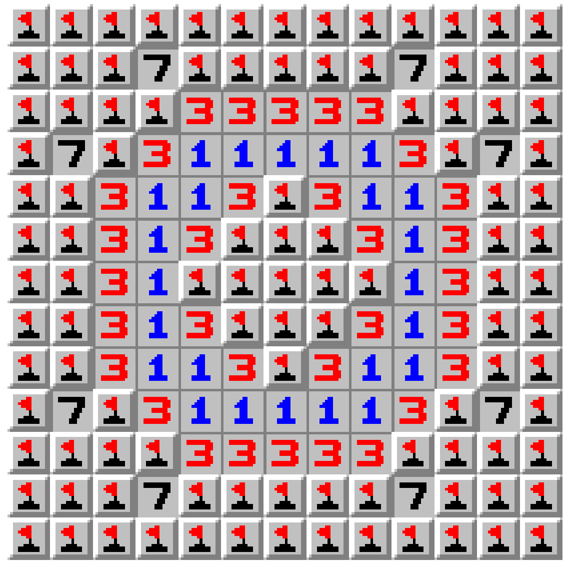
5.  6.  7.  8. 

9.  10.  11. 

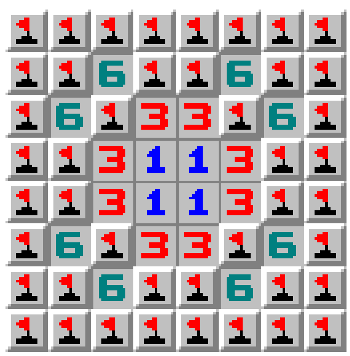
12.  13.  14. 



15.

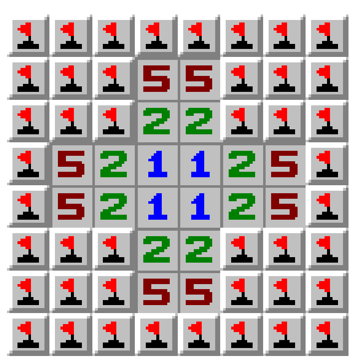


16.

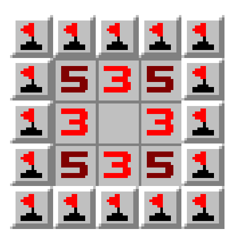


17.

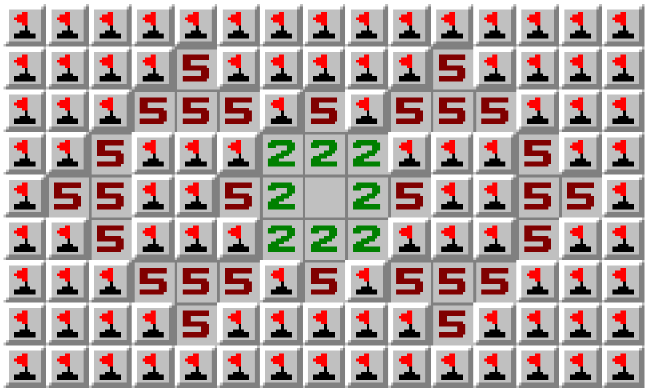
18.

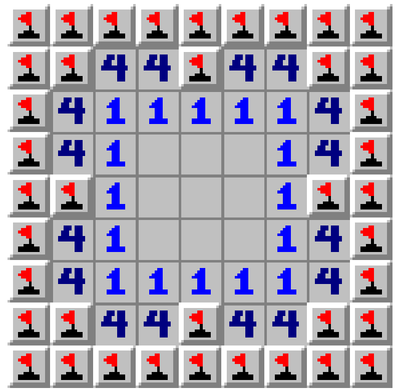


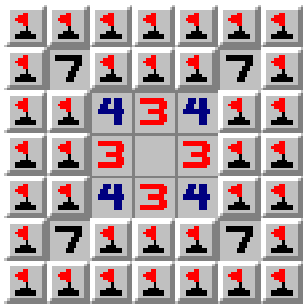
19.

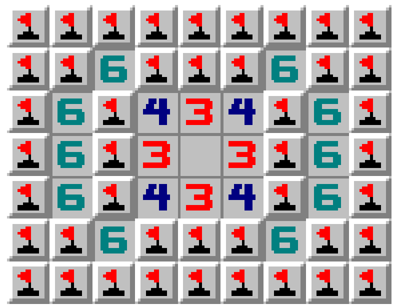


20.

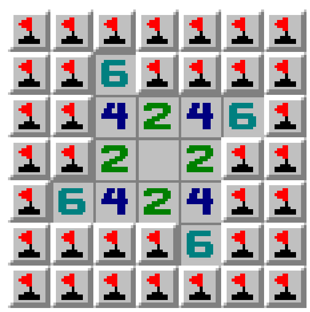
21. 

22. 

23. 

24. 

25. 

26. 

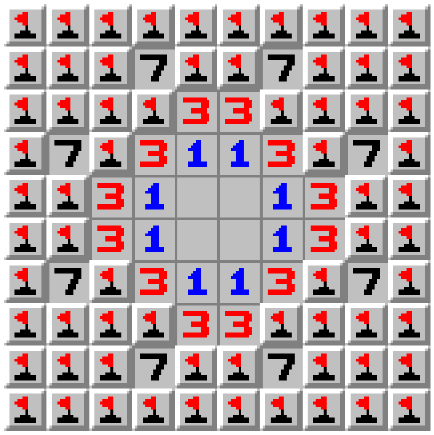
27.  28. 

Figure 2-1: Base cases of mine-bordered boards.

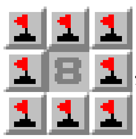
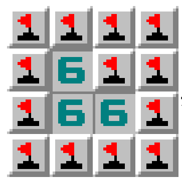
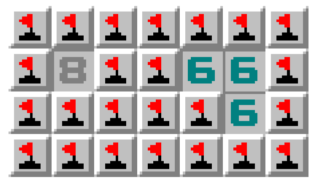
2.2.2 Union Theorem

Theorem 1. *(Mine-Bordered Boards Union Theorem) If sets S and T are both ranges of valid mine-bordered boards, then $S \cup T$ is also an range of a mine-bordered board; in other words, ranges of valid mine-bordered boards are closed under union.*

Proof. Let B_S be a mine-bordered board with range S and B_T be a mine-bordered board with range T. Define $w(B_S)$ and $h(B_S)$ as the width and height of B_S respectively, and $w(B_T)$ and $h(B_T)$ as the width and height of B_T respectively. Construct a new board B_U with width $w(B_S)+w(B_T)$ and height the max of $h(B_S)$ and $h(B_T)$. Copy B_S into the first $w(B_S)$ columns and first $h(B_S)$ rows of B_U, and copy B_T into the last $w(B_T)$ columns and first $h(B_T)$ rows of B_U. Fill all squares occupied neither by the copy of B_S nor by the copy of B_T with mines. This resulting board B_U is valid because all internal cells of the copies of B_S and B_T remain valid due to bordering the exact same set of cells as in the valid boards from which they were copied, and all other cells are mines, which are valid regardless of what values are in cells they border. The range of B_U is the union of the ranges of B_S and B_T because a number appears in at least one cell of B_U if and only if it appears in a cell of at least one of B_S and B_T: every number in B_S or B_T is in B_U as the entire boards are copied over, and every number not in B_S or B_T is not in B_U because all other cells of B_U contain mines, not numbers. □

This theorem alone, along with our base-case mine-bordered boards in Section 2.2.1, proves the validity of a majority of the 512 ranges.

Example 1. *With B_S as base case 1 (with range $\{8\}$) and B_T as base case 3 (with range $\{6\}$), we construct B_U with range $\{6\} \cup \{8\} = \{6, 8\}$.*

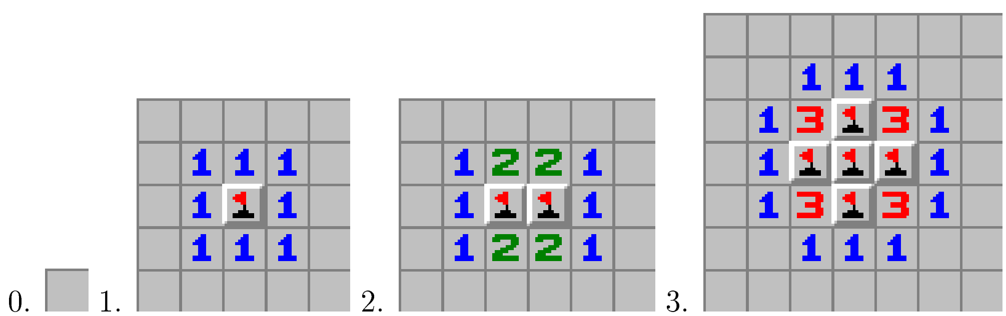
$B_S =$ , $B_T =$ , $B_U =$ 

2.3 Zero-Bordered Boards

Definition 4. *A zero-bordered board is a board where every cell on each of its edges contains a zero.*

2.3.1 Base Cases

Figure 2-2 displays the six base cases of zero-bordered boards.



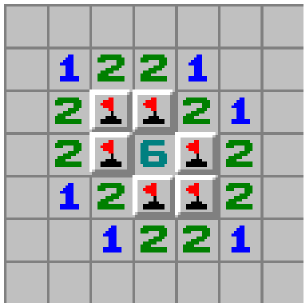
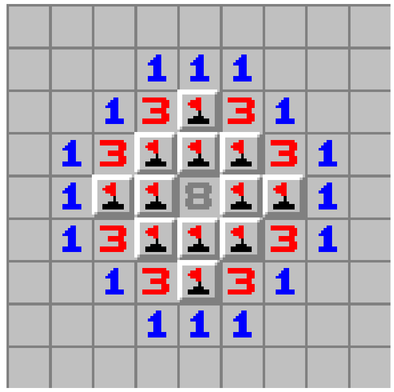
4.  5. 

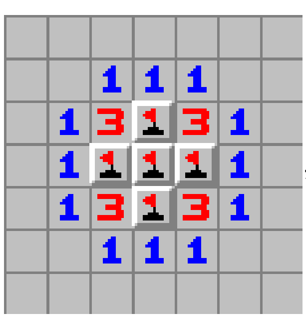
Figure 2-2: Base cases of zero-bordered boards.

2.3.2 Union Theorem

Theorem 2. *(Zero-Bordered Boards Union Theorem) ranges of valid zero-bordered boards are closed under union.*

Proof. Follow the exact same construction as is used to prove the Mine-Bordered Boards Union Theorem (Section 1), but replace mines with zeroes. □

Example 2. *With* B_S *as base case 3 (with range* $\{0,1,3\}$*) and* B_T *as base case 2 (with range* $\{0,1,2\}$*), we construct* B_U *with range* $\{0,1,3\} \cup \{0,1,2\} = \{0,1,2,3\}$.

$B_S =$  $, B_T =$  ,

$B_U =$ 

2.4 Wall Boards

2.4.1 Union Theorem

Theorem 3. *(Wall Union Theorem) If set S is an range of a valid mine-bordered board, and set T is an range of a valid zero-bordered board, then $S \cup T \cup \{2, 3\}$ is a valid range.*

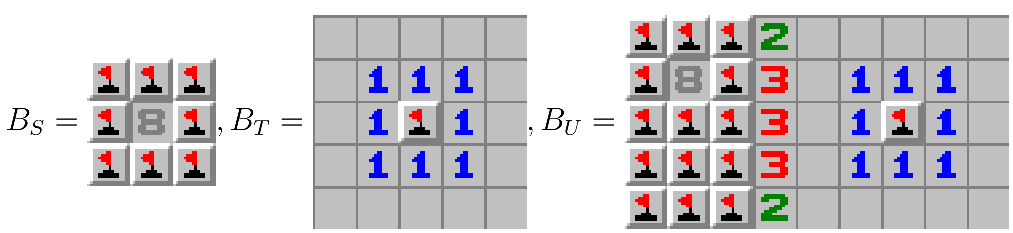
Proof. Let B_S be a mine-bordered board with range S and B_T be a zero-bordered board with range T. Define $w(B_S)$ and $h(B_S)$ as the width and height of B_S respectively, and $w(B_T)$ and $h(B_T)$ as the width and height of B_T respectively. Construct a new board B_U with width $w(B_S) + w(B_T) + 1$ and height the max of $h(B_S)$ and $h(B_T)$. Copy B_S into the first $w(B_S)$ columns and first $h(B_S)$ rows of B_U, and copy B_T into the last $w(B_T)$ columns and first $h(B_T)$ rows of B_U. Fill the remaining column in the middle with a 2 at the top and bottom cells and a 3 in all other cells. Fill all cells left of these two columns unoccupied by the copy of B_S with mines, and fill all cells right of these two columns unoccupied by the copy of B_T with zeros. In the resulting board, internal cells of copies of B_S and B_T remain valid, all zeros right of the middle columns are valid because the B_T internal cells that they border have not changed and no other cells bordering cells right of the middle columns have mines, and the 2s and 3s in the right middle column are valid because they represent the number of mines in cells bordering their cells in the column to the left, whereas no cells in the column to the right contain mines as they are either cells on the border of zero-bordered board B_T or filled-in zeros. All remaining cells contain mines, which are valid regardless of what cells they border. The range of B_U is $S \cup T \cup \{2, 3\}$ because all numbers outside the copies of B_S and B_T are either the 2s and 3s in the

range	MBB	ZBB
$\{0,2,3,4,8\}$	$\{3,4\} \cup \{8\}$	$\{0\}$
$\{0,2,3,4\}$	$\{3,4\}$	$\{0\}$
$\{0,2,3,6,7,8\}$	$\{6\} \cup \{7\} \cup \{8\}$	$\{0\}$
$\{0,2,3,7,8\}$	$\{7\} \cup \{8\}$	$\{0\}$
$\{0,2,3,6,8\}$	$\{6\} \cup \{8\}$	$\{0\}$
$\{0,2,3,8\}$	$\{8\}$	$\{0\}$
$\{0,2,3,6,7\}$	$\{6\} \cup \{7\}$	$\{0\}$
$\{0,2,3,7\}$	$\{7\}$	$\{0\}$
$\{0,2,3,6\}$	$\{6\}$	$\{0\}$
$\{0,2,3\}$	$\{\}$	$\{0\}$
$\{2,3,8\}$	$\{8\}$	$\{\}$
$\{2,3\}$	$\{\}$	$\{\}$

Table 2.2: Corresponding MBBs and ZBBs for ranges proven possible with the Wall Union Theorem (Section 2.4.1).

right middle column or 0s added right of the middle columns, which does not affect the union because B_T as a zero-bordered board necessarily already contains zeros. □

Example 3. *With B_S as MBB base case 1 (with range $\{8\}$) and B_T as ZBB base case 1 (with range $\{0,1\}$), we construct B_U with range $\{8\} \cup \{0,1\} \cup \{2,3\} = \{0,1,2,3,8\}$.*

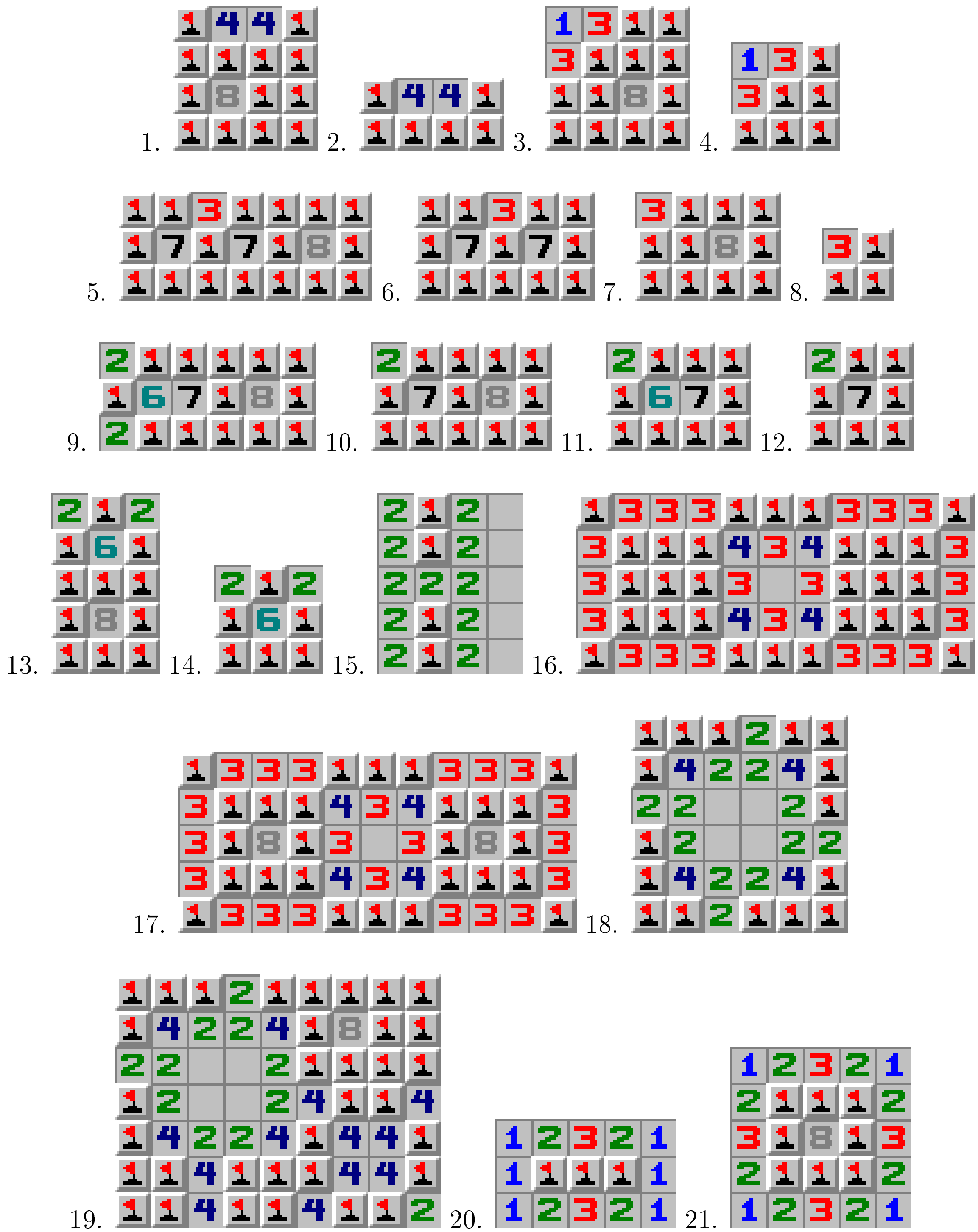


2.4.2 Construction of Wall Cases

There are twelve ranges proved valid using the Wall Union Theorem. Table 2.2 contains the corresponding MBBs and ZBBs the Wall Union Theorem is applied to to produce each of these twelve ranges.

2.5 Individual Cases

The boards in Figure 2-3 are presented as individual cases, used only for proving their own ranges possible.



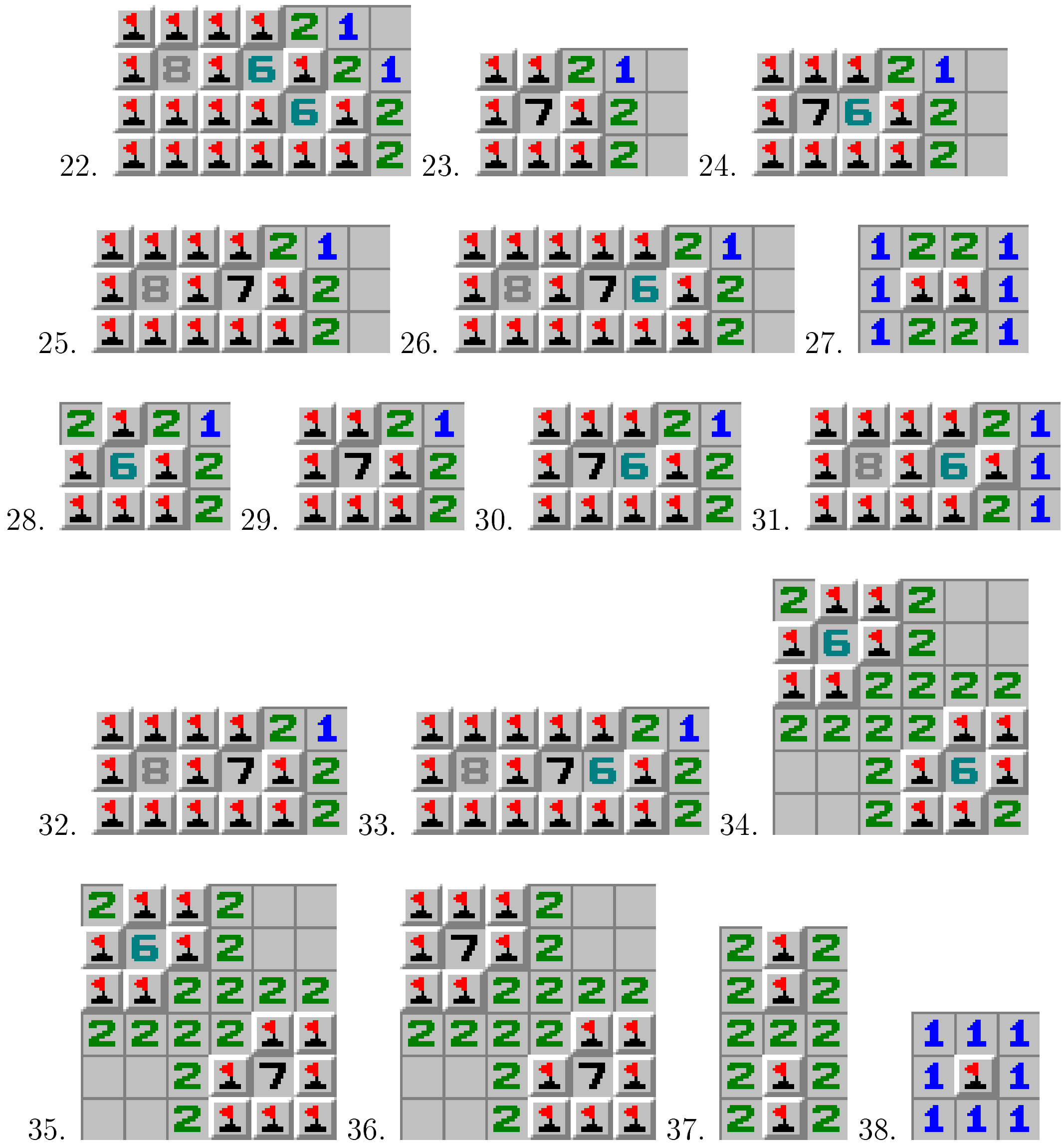


Figure 2-3: Boards for individually-proven ranges.

2.6 Gap Impossibility Theorems

Lemma 1. *Two horizontally or vertically adjacent cells on a valid Minesweeper board cannot contain numbers differing by more than 3.*

Proof. If two cells are horizontally or vertically adjacent, we already know that at least one of the eight surrounding cells for both of these cells does not contain a mine, that is, the other cell. Of the seven remaining cells, four are shared between the two cells in coverage, and thus whether any of them contain a mine does not affect the difference between the two numbers. Hence, as there are only three cells on a side for which their contents can affect the difference between the two numbers, and each of these cells can only alter the difference by 1, the maximum difference between two horizontally or vertically adjacent cells is 3. □

Lemma 2. *Two diagonally adjacent cells on a valid Minesweeper board cannot contain numbers differing by more than 5.*

Proof. Use the same proof as in Lemma 1, except that of the seven remaining cells only two are shared in the coverages of the original cells, and thus five cells are leftover. □

Lemma 3. *If cell A is not adjacent to cell B, then there are at least three cells adjacent to cell A that are closer to cell B (by straight-line distance between cell centers) than cell A is, as long as the line of cells from cell A to cell B does not follow an edge of the board.*

Proof. We'll prove this for two cases: cell A being on the same horizontal or vertical line as cell B, and cell A not being on the same horizontal or vertical line as cell B.

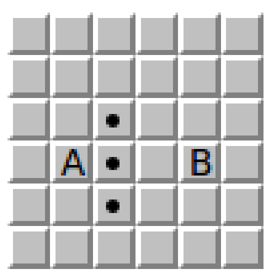


Figure 2-4: Three cells closer to B than A is when A and B are horizontally aligned.

If cell A is on the same horizontal or vertical line as cell B, then assume without loss of generality that this is a horizontal line with A on the left and B on the right. (Otherwise, rotate the board for the rest of this argument.) Consider the cell immediately to the right of cell A, as well as the two cells immediately above and below this cell. These cells exist, because we assumed the line of cells from cell A to cell B does not follow an edge of the board, which implies that there is at least one column or row of cells on either side. As all three of these cells are adjacent to cell A, they can't be cell B, as that would violate the assumption that A and B are not adjacent. But all three of these cells are closer to cell B than cell A is: the cell on the line between A and B, and the other two cells by the triangle inequality, since the Manhattan distance of these cells to B is the same as the straight-line distance from A to B.

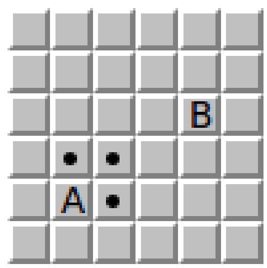


Figure 2-5: Three cells closer to B than A is when A and B are not horizontally or vertically aligned.

If cell A is not on the same horizontal or vertical line as cell B, then assume without loss of generality that B is to the upper-right of A. (Once again, rotate the board for the rest of this argument otherwise.) Consider the squares immediately above, immediately to the right, and immediately to the upper-right of A. Once again, all three of these cells are adjacent to A, so they can't be B, and all three are closer to B than A is. □

Theorem 4. *(2-8 Gap Impossibility Theorem) A valid Minesweeper board cannot contain both a number 2 or lower and an 8 without also containing a number strictly between 2 and 8.*

Proof. Suppose there existed a valid Minesweeper board containing at least one number 2 or lower, at least one 8, and no numbers between 2 and 8. Consider the $(\leq 2, 8)$ pair with the smallest distance between them (in the case of a tie, consider any pair in the tie), and let c_1 be the cell containing the number ≤ 2 and let c_2 be the cell containing the 8. If c_1 and c_2 are on the same vertical or horizontal line, then they do not follow an edge of the board, as there can't be 8s on an edge of a board. Cells c_1 and c_2 are not adjacent, because by Lemmas 1 and 2 adjacent cells cannot contain numbers differing by more than 5. But then by Lemma 3 there are three cells adjacent to c_1 that are closer to c_2 than c_1 is in this smallest-distance pair. If any of these cells contained a number 2 or less or an 8, it would violate the fact that we were previously considering the smallest-distance pair. If any of these cells contained a number strictly between 2 and 8, it would contradict our original assumption that the board had no numbers strictly between 2 and 8. Thus, none of these cells can contain numbers, but if all three of these cells contained mines, it would contradict c_1 containing a number ≤ 2. Thus, any case of the values of these cells results in contradiction. □

Theorem 5. *(1-5 Gap Impossibility Theorem) A valid Minesweeper board cannot contain both a number 1 or lower and a number 5 or higher without also containing a number strictly between 1 and 5.*

Proof. Suppose there existed a valid Minesweeper board containing at least one number 1 or lower, at least one number 5 or higher, and no numbers strictly between 1 and 5. Consider the $(\leq 1, \geq 5)$ pair with the smallest distance between them (in the case of a tie, consider any pair in the tie), and let c_1 be the cell containing the number ≤ 1 and let c_2 be the cell containing the number ≥ 5. We'll consider two cases: c_1 and c_2 are adjacent, and c_1 and c_2 are not adjacent.

If c_1 and c_2 are adjacent, then they must be diagonally adjacent, as the difference between the two cells is more than 3 and Lemma 1 prohibits vertical and horizontal adjacencies between cells containing such numbers. Consider the two cells adjacent to both c_1 and c_2. If one of these two cells contains a number strictly between c_1's and c_2's numbers, it is in violation of the assumption. If its number is at most c_1's number, it is in violation of Lemma 1 with c_2. If its number is at least c_2's number, it is in violation of Lemma 1 with c_1. Thus, these squares can't contain numbers.

But both containing mines would contradict c_1's number being at most 1. So every situation where the cells are adjacent leads to contradiction.

Suppose this pair of cells is not adjacent. This board contains at least two rows and at least two columns, as otherwise the board cannot contain a 5. If c_1 and c_2 are on the same row or column and this row or column happens to follow an edge of the board, then there are two cells adjacent to c_1 that are closer to c_2 than c_1 (since the board has at least two rows and at least two columns, there must be a respective row or column on at least one side). Otherwise, by Lemma 3 there are three cells adjacent to c_1 that are closer to c_2 than c_1 is in this smallest-distance pair. In either case, if any of these cells contained a number ≤ 1 or ≥ 5, that would violate these being the smallest-distance pair. If any of these cells contained a number strictly between 1 and 5, it would violate the assumption. Thus, none of these cells can contain numbers, but if all two or three of these cells contained mines, it would contradict c_1 containing a number ≤ 1.

As each possibility in these cases leads to contradiction, there must not be such valid Minesweeper boards. □

Lemma 4. *A cell containing a 0 cannot be horizontally or vertically adjacent to either a number 4 or above or a mine.*

Proof. Lemma 1 disallows 0s to be horizontally or vertically adjacent to numbers 4 or larger, as their difference will be greater than 3. A 0 can't be adjacent to a mine either, as '0' indicates that there are no mines in the cells adjacent to where the 0 resides. □

Theorem 6. *(0-4 Gap Impossibility Theorem) A valid Minesweeper board cannot contain both a 0 and a number 4 or higher without also containing a number strictly between 0 and 4.*

Proof. Such a board would need to contain at least one 0. Consider any row that contains a 0. All cells in this row must contain 0s, as the presence of a number 4 or above or mine violates Lemma 4. For the same reason, all other rows also need to contain all 0s. Thus, the board consists only of 0s, violating the assumption that it contained a cell with a number 4 or above. □

Theorem 7. *(0-3 Gap Impossibility Theorem) A valid Minesweeper board cannot contain both a 0 and a 3 without also containing at least one 1, 2, 4, or 5.*

Proof. Assume to the contrary that such a valid Minesweeper board exists. The board must contain a 0. Consider the contiguous mass of 0s that some 0 is part of. As the board also contains at least one 3, this contiguous mass of 0s does not span the whole board. Thus there is at least one cell adjacent to a 0 that is not a 0. This cell cannot contain a number ≥ 6, by Lemmas 1 and 2. This cell can't be a mine, as it is adjacent to a 0. Thus, it must be a 3. Such cells containing 3s cannot be on the corner or edge of the board, as in such locations adjacency to a 0 means placing 3 mines in the squares adjacent to the 3 will necessarily place at least one adjacent to the 0. Thus, any contiguous mass of 0s either includes the entire border of the board (in which case all non-0 squares form an "island" of cells in the interior) or does not touch the border of the board anywhere.

In both cases, as cells containing numbers ≥ 6 or mines can't be adjacent to cells with 0s, the entire set of cells bordering at least one 0 must all have 3s.

In the former case, consider the non-0 cells, whose outer border consists entirely of 3s. This outer boundary must contain some cell that touches both a vertically adjacent 0 and a horizontally adjacent 0 (consider a convex corner of the region). In this case, the 3 cannot be satisfied, as there is only one square adjacent to the 3 that isn't adjacent to a 0.

In the latter case, where the contiguous mass of 0s doesn't touch the border, the 0s are surrounded by 3s. If the mass of 0s forms a concave region, then there is at least one 3 that touches both a vertically adjacent 0 and a horizontally adjacent 0 (consider a concave corner of the region). As in the former case, this 3 cannot be satisfied. If the mass of 0s does not form a concave region, then it is a rectangle. If this rectangle is surrounded by 3s, it means that on all sides of this rectangle, the next row or column of cells beyond the row or column of 3s are all mines, in order to satisfy the 3s horizontally or vertically adjacent to the edges of this rectangle. Doing so violates the 3s in the corners of these regions, as they are now adjacent to at least four mines.

Thus, all possibilities lead to contradiction. □

Chapter 3

Complexity of the Consistency Problem in Modified Minesweeper Games

Definition 5. *S-Minesweeper consistency is the Minesweeper consistency problem where only numbers in the set S are allowed to be entries of non-mine cells in the input and completed boards.*

In Table 3.1, each cell corresponding to a subset S of $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ displays whether S-Minesweeper consistency is in this section proved in P or NP-complete, or is still an open question.

Some of the open problems are equivalent to each other. We use "open(k)" to denote cases that are all in P if and only if any individual case is in P, and NP-complete if and only if any individual case is NP-complete. Different numbers k denote different such clusters of equivalent cases.

3.1 Easy Cases: {}-Minesweeper, {0}-Minesweeper, and {8}-Minesweeper

Theorem 8. *$\{\}$-Minesweeper consistency $\in P$.*

∪	{6, 7, 8}	{7, 8}	{6, 8}	{8}	{6, 7}	{7}	{6}	{}
{0, 1, 2, 3, 4, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 3, 4, 5}	open	open	open	open	open	open	open	open
{0, 2, 3, 4, 5}	open	open	open	open	open	open	open	open
{2, 3, 4, 5}	open	open	open	open	open	open	open	open
{0, 1, 3, 4, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 3, 4, 5}	open	open	open	open	open	open	open	open
{0, 3, 4, 5}	open	open	open	open	open	open	open	open
{3, 4, 5}	open	open	open	open	open	open	open	open
{0, 1, 2, 4, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 4, 5}	open	open	open	open	open	open	open	open
{0, 2, 4, 5}	open	open	open	open	open	open	open	open
{2, 4, 5}	open	open	open	open	open	open	open	open
{0, 1, 4, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 4, 5}	open	open	open	open	open	open	open	open
{0, 4, 5}	open(1)	open(2)	open(3)	open(4)	open(5)	open(6)	open(7)	open(8)
{4, 5}	open(1)	open(2)	open(3)	open(4)	open(5)	open(6)	open(7)	open(8)
{0, 1, 2, 3, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 3, 5}	open	open	open	open	open	open	open	open
{0, 2, 3, 5}	open	open	open	open	open	open	open	open
{2, 3, 5}	open	open	open	open	open	open	open	open
{0, 1, 3, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 3, 5}	open	open	open	open	open	open	open	open
{0, 3, 5}	open	open	open	open	open	open	open	open
{3, 5}	open	open	open	open	open	open	open	open
{0, 1, 2, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 5}	open	open	open	open	open	open	open	open
{0, 2, 5}	open	open	open	open	open	open	open	open
{2, 5}	open	open	open	open	open	open	open	open
{0, 1, 5}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 5}	open(9)	open(10)	open(11)	open(12)	open(13)	open(14)	open(15)	open(16)
{0, 5}	open(9)	open(10)	open(11)	open(12)	open(13)	open(14)	open(15)	open(16)
{5}	open(9)	open(10)	open(11)	open(12)	open(13)	open(14)	open(15)	open(16)
{0, 1, 2, 3, 4}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 3, 4}	open	open	open	open	open	open	open	open
{0, 2, 3, 4}	open	open	open	open	open	open	open	open
{2, 3, 4}	open	open	open	open	open	open	open	open
{0, 1, 3, 4}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 3, 4}	open	open	open	open	open	open	open	open
{0, 3, 4}	open	open	open	open	open	open	open	open
{3, 4}	open	open	open	open	open	open	open	open
{0, 1, 2, 4}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 4}	open	open	open	open	open	open	open	open
{0, 2, 4}	open	open	open	open	open	open	open	open
{2, 4}	open	open	open	open	open	open	open	open
{0, 1, 4}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 4}	open	open	open	open	open	open	open	open
{0, 4}	open(17)	open(18)	open(19)	open(20)	open(21)	open(22)	open(23)	open(24)
{4}	open(17)	open(18)	open(19)	open(20)	open(21)	open(22)	open(23)	open(24)
{0, 1, 2, 3}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2, 3}	open	open	open	open	open	open	open	open
{0, 2, 3}	open	open	open	open	open	open	open	open
{2, 3}	open	open	open	open	open	open	open	open
{0, 1, 3}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 3}	open	open	open	open	open	open	open	open
{0, 3}	open(25)	open(26)	open(27)	open(28)	open(29)	open(30)	open(31)	open(32)
{3}	open(25)	open(26)	open(27)	open(28)	open(29)	open(30)	open(31)	open(32)
{0, 1, 2}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1, 2}	open	open	open	open(33)	open	open	open	open(33)
{0, 2}	open	open	open	open(34)	open	open	open	open(34)
{2}	open	open	open	open(35)	open	open	open	open(35)
{0, 1}	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete	NP-complete
{1}	open(36)	open(37)	open(38)	P	open(39)	open(40)	open(41)	P
{0}	open(36)	open(37)	open(38)	P	open(39)	open(40)	open(41)	P
{}	open(36)	open(37)	open(38)	P	open(39)	open(40)	open(41)	P

Table 3.1: Complexity of S-Minesweeper consistency for the 512 possibilities of S.

Proof. All instances of $\{\}$-Minesweeper are consistent with a board entirely filled with mines, as no numbers are allowed. A board with only mines is always valid because mines do not require any restrictions on adjacent cells for validity. □

Theorem 9. *$\{0\}$-Minesweeper consistency $\in P$.*

Proof. All instances of $\{0\}$-Minesweeper are consistent with a board with no mines, as the presence of even a single mine in a board not filled with mines will necessitate some number greater than 0. Thus, Minesweeper consistency with the set $\{0\}$ is merely a matter of seeing whether the given partial board has any mines whatsoever. □

Theorem 10. *$\{8\}$-Minesweeper consistency $\in P$.*

Proof. Checking Minesweeper consistency on a board with only 8s is simply a matter of seeing whether there are adjacent 8s or 8s on the boundary. If so, the board is not consistent, and if not, the board is consistent, as one can simply assign all unassigned squares to have mines and satisfy each 8. □

3.2 $\{1\}$-Minesweeper

Lemma 5. *A Minesweeper board with only* 1*s and mines cannot have mines in the* $3n$*th row or column from any side.*

Proof. We will show that such a board cannot have a mine in the third row from the edge of a board, and then show that such a board cannot have a mine in the third row or column from a row or column known to have no mines.

For any mine in a board with only 1s and mines, none of the surrounding cells can contain mines. If there was a contiguous mass of more than one mine, then either this contiguous mass takes up the entire board, which would contradict the board having 1s, or this contiguous mass has numbered cells on its boundary, of which there must be some number higher than 1. Consider any mine on a third row or column from an edge. All cells adjacent to the mine's cells must thus have 1s, including all three (possibly two, if on the edge on the other axis) cells on the second row or column from the edge adjacent to this cell. Now, consider the cell on the first row or column from the edge but on the same column or row, respectively, as the cell containing a mine. This cell can contain neither a mine (as that would contradict the 1s in the second row or column), nor a 1 (as that would imply either square next to it on the first row or column contains a mine, contradicting some 1s on the second row or column), thus contradicting that this board contains only 1s and mines. Thus, a Minesweeper board with only 1s and mines cannot have mines in the 3rd row or column.

Now consider a mine three rows or columns from a row or column known to not have mines. Use the same process as for a row or column 3 rows or columns from the edge, but in analyzing the possibility of a 1 on the first row or column, the adjacent mine cannot be in the previous row or column (where, in the previous case, the edge was) because that would violate the given that the original row or column was the third row or column from a row or column known to have no mines.

Thus, a Minesweeper board with only 1s and mines cannot have mines in the $3n$th row or column from any side. □

Theorem 11. $\{1\}$*-Minesweeper consistency* $\in P$.

Proof. We will consider three cases:

1. Both the width and the height of the board are not $2 \pmod 3$.

2. One of the width and the height of the board is $2 \pmod 3$; the other isn't.

3. Both the width and the height of the board are $2 \pmod 3$.

If both the width and the height of the board are not $2 \pmod 3$, then the above lemma eliminates the possibilities of mines in two different sets of rows when applied from opposite sides of a board. Thus, the only possible cells in which mines can reside are two rows apart from the next possible cell in the same column, and two columns apart from the next possible cell in the same row. Also, these must all contain mines: if any of these cells is not a mine, there can't be 1s in adjacent cells because the next possible location of a mine is 3 rows or 3 columns away. Evaluating consistency a board is then merely the process of checking that each such location in an input board has a mine, and each other location in an input board has a 1.

Suppose one of the width and the height of the board are not 2 (mod 3). Assume without loss of generality that it's the width that isn't 2 (mod 3). Then, the rows eliminated from each end are different sets, so rows that can have mines are 3 rows apart. Thus, the board consists of groups of 2 adjacent cells that can have mines, of which exactly one must in fact be a mine: if neither in a group is a mine, then with no mines adjacent, the cells would contain 0s, which is a contradiction, and if both are mines, some adjacent cell will have a 2, also a contradiction. In addition, all groups in the same row must have the mine in the same corresponding location (either the left cell or the right cell of the group): if for any two consecutive groups in the row the left group has the left cell with a mine and the right group has the right cell with a mine, then the cell between the groups cannot have a mine and will have no mines adjacent, thus being a 0, and if the left group has the right cell with a mine and the right group has the left cell with a mine, then the cell between the groups cannot have a mine but is adjacent to at least two mines. Hence, a polynomial-time algorithm to check any input board for consistency simply checks that cells in the $3n$th row or column from any side do not have mines, and then checks to make sure no row or column (depending on which of width and height isn't 2 (mod 3)) has both a group with a left-cell 1 and a group with a right-cell 1.

Suppose both the width and the height of the board are 2 (mod 3). Then, both only every third row and every third column is guaranteed to not have mines. This means that mines can now either line up in rows or in columns, but still in the consistent position across groups, due to the same constraints as in the previous case. Hence, a polynomial-time algorithm for deciding consistency in this case would, after verifying no mines in the prohibited rows and columns, assume mines line up in rows and then check the conditions in the previous case, then assume mines line up in columns and check the corresponding conditions, and accept if either choice is consistent.

Since in all cases consistency can be decided in polynomial time, $\{1\}$-Minesweeper consistency $\in P$. □

3.3 {0,1}-Minesweeper

Theorem 12. *$\{0,1\}$-Minesweeper consistency is NP-complete.*

Proof. $\{0,1\}$-Minesweeper consistency is in NP. As with the general Minesweeper consistency problem, consider a completed board consistent with the input board itself as a certificate. This certificate can be verified in polynomial time by simply checking that each number does in fact reflect the number of mines in adjacent cells, and that the entry in each cell corresponding to a filled-in cell in the input board is in fact the same entry as in the input board.

We will prove $\{0,1\}$-Minesweeper NP-hard by polynomial-time reduction from Planar Positive Rectilinear 1-in-3SAT, an NP-complete problem [4]. Planar Positive Rectilinear 1-in-3SAT instances are instances of 1-in-3SAT where literals only affirm (not negate) variables, and a graph consisting of one node for each variable and clause can have all edges from a variable to all clauses it appears in not intersect and be always parallel to the horizontal or vertical axis (allowing right-angle turns).

Using this planar rectilinear form of a 1-in-3SAT problem, create wires (Figure 3-1) for each edge in the graph. Represent setting the variable to true in the SAT instance with setting the undecided cell between the first two 1s of the wire (from the variable-node end) to a mine, and represent setting the variable to false in the SAT instance with setting the undecided cell between the second and third 1s of the wire to a mine. Notice that since every second cell along the wire contains a 1, and 0s two rows or columns from the wire prevent setting adjacent cells off the path of the wire to be mines, satisfying the 1s requires every second undecided cell along the wire to be a mine, so the setting of the first mine determines the locations of the mines down the wire, and thus a boolean value can be propagated indefinitely.

Construct turn gadgets (Figure 3-2) at each right-angle turn in the edges. For each additional clause a variable appears in, use a split gadget (Figure 3-3) to copy the value of a variable along a wire to two wires. (Each variable thus comes with

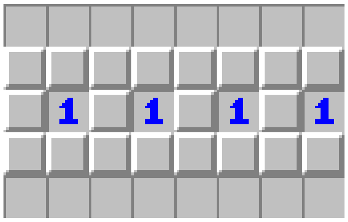


Figure 3-1: wire

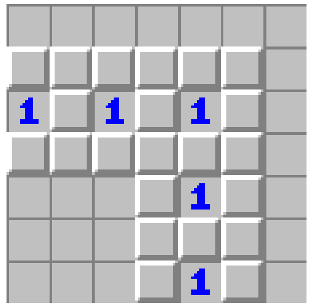


Figure 3-2: turn gadget

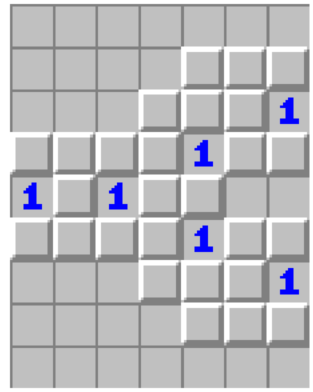


Figure 3-3: split gadget

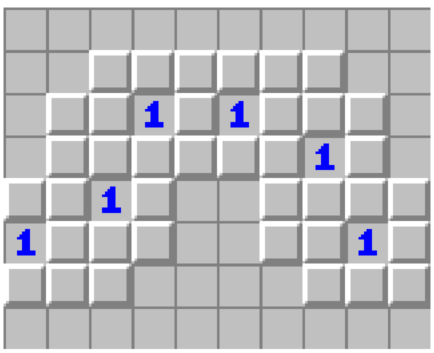


Figure 3-4: phase shift gadget

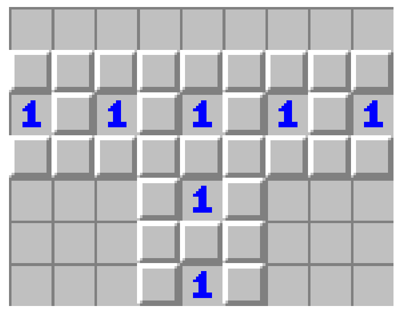


Figure 3-5: clause gadget

$k-1$ split gadgets if the variable appears k times in the problem instance.) If the wire ever needs to shift phase, construct a phase shift gadget (Figure 3-4). Because the SAT variant we are reducing from is planar, we do not need a gadget for crossing wires.

Finally, for each clause in the Planar-Positive-Rectilinear-1-in-3SAT instance, construct a clause gadget (Figure 3-5), such that the 1 at the junction touches cells that would contain mines if the variable represented by the wire is set to true. (If a mine in the cell would have represented setting the wire to false, incorporate a phase shift in the wire so that the mine represents setting the wire to true.)

If the original SAT instance is satisfiable, then the truth values assigned to the variables, when reflected in this $\{0,1\}$-Minesweeper consistency instance, would be such that each clause has precisely one cell next to the 1 at its junction containing a mine, that is, the path relating to a variable set to true. If the original SAT instance is unsatisfiable and no violations occurred along the wires themselves (and thus truth values are propagated), then there will exist some clause for which the number of the three cells around the 1 at its junction that are mines is different from 1, thus being inconsistent with the input board, violating the 1 at the junction.

Each of these gadgets contains a constant number of cells, and each only needs to be implemented a polynomial number of times to represent corresponding elements of the 1-in-3SAT instance graph (only one initial wire and as many split gadgets and instances of a variable are needed for variables, and only one clause gadget is necessary per clause), so this reduction takes polynomial time. □

3.4 Implied Results

Whenever S-Minesweeper consistency is proved NP-complete, so is T-Minesweeper consistency for all $T \supset S$, since all instances of S-Minesweeper consistency are also instances of T-Minesweeper consistency, providing the identity reduction for NP-hardness, and all instances are in NP via a simple consistency check. With as small an S as $\{0, 1\}$, this allows us to conclude many other NP-completenesses instantaneously: each of the 127 proper supersets of $\{0, 1\}$ in $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$.

The same immediate implications hold for membership in P and $T \subset S$, but this implication is unlikely to yield as many actual results, particularly given the above, which suggests that most of these restricted-set instances of Minesweeper consistency are NP-complete.

3.5 Relating Problems Using Ranges

If S is an invalid range, then that means there are no valid completed Minesweeper boards for which S is the exact set of numbers appearing at least once on the board, which means the question of whether S-Minesweeper consistency is NP-complete can be rephrased in terms of the NP-completenesses of proper subsets of S.

Lemma 6. *Suppose S is an invalid range. If T-Minesweeper consistency is in P for all*

$T \in \{S \backslash \{a\} \mid a \in S\}$, then so is S-Minesweeper consistency, and if at least one such T-Minesweeper consistency is NP-complete, then so is S-Minesweeper consistency.

Proof. If S is an invalid range, then no valid completed Minesweeper board has exactly the set of numbers in S appearing on the board. Thus, the set of numbers that actually appear is some subset of S. Every proper subset of S is a subset of at least one T for $T \in \{S \backslash \{a\} \mid a \in S\}$. Thus, if T-Minesweeper consistency is in P for all such T, then S-Minesweeper consistency is as well, as one could solve S-Minesweeper consistency by solving each valid T-Minesweeper consistency problem. If at least one such T-Minesweeper consistency problem is NP-complete, then since all instances of this T-Minesweeper consistency problem are also instances of S-Minesweeper consistency, then S-Minesweeper consistency is NP-complete. □

Theorem 13. $\{0, 8\}$-*Minesweeper consistency is* $\in P$.

Proof. $\{0, 8\}$ is an invalid range, so it is $\in P$ if both $\{0\}$-Minesweeper consistency and $\{8\}$-Minesweeper consistency are in P. They have both been proved in P, so $\{0, 8\}$-Minesweeper consistency $\in P$. □

Theorem 14. $\{0, 4\}$-*Minesweeper consistency is* $\in P$ *if and only if* $\{4\}$-*Minesweeper consistency is* $\in P$.

Proof. $\{0, 4\}$ is an invalid range, so it is $\in P$ if both $\{0\}$-Minesweeper consistency and $\{4\}$-Minesweeper consistency are in P. $\{0\}$-Minesweeper consistency is $\in P$, so $\{0, 4\}$-Minesweeper consistency is $\in P$ if $\{4\}$-Minesweeper consistency is, and NP-complete if $\{4\}$-Minesweeper consistency is. □

These relations allow us to form the 41 groups of problems whose complexities depend on each others' in the chart at the beginning of this section, using the impossibilities of ranges.

Chapter 4

Further Questions

Many further variants of the Minesweeper game can be analyzed for combinatorial properties and complexity. In the above work, we only analyze finite rectangular flat Minesweeper boards, and even our analysis of this is not complete.

4.1 Still-Open Restricted-Range Minesweeper Problems

There are still 3 of the 512 ranges for which we have not determined validity. Each of these three sets of numbers suffer from massive gaps that suggest them impossible, but we have not found a concrete proof. Similar cases are arguably surprising: $\{0, 2, 7\}$, for instance, does not seem intuitively possible.

As for complexity, given that even just 0 and 1 are enough for an NP-complete variant of Minesweeper, it is reasonable to expect most of the 512 subsets of numbers correspond to NP-complete variants. It intuitively makes sense that most sets of numbers that are valid ranges via the MBB, ZBB, and WALL Union Theorems would correspond to NP-complete Minesweeper consistency variants, as the corresponding constructions are very flexible and have many structures that are easily extended. Many individually proved cases, though, are restrictive enough situations that it may very well be there are insufficiently different boards to pose too much of a challenge to checking consistency. It may even be that for some of the subsets (e.g. $\{0, 2, 6\}$) the number of valid boards with that range is finite, given how much some of the individual cases' validity depend on edge structure. It would be interesting to characterize which valid ranges have only finitely many satisfying boards.

Versions of Minesweeper can be viewed as boolean satisfiability (SAT) problems with specifications on the number of evaluating-true literals in a clause and a constraining distance metric. For instance, $\{1\}$-Minesweeper on a rectangular 2-D board can be viewed as Positive 1-in-8SAT where literals are only allowed in a clause if their mapping onto a 2-dimensional Euclidean metric space is within a distance of where the clause is mapped; this imposes constraints like that a variable can appear at most 8 times in the problem statement. Unfortunately, when the set of numbers permitted on a Minesweeper board expands, mapping to SAT problems becomes more difficult as different clauses in a direct mapping will have different constraints, and thus not adhere to a consistent version of SAT. Nevertheless, analysis of versions of SAT and Minesweeper could likely help with analysis of versions of the other.

4.2 Post- versus Pre- Restricted Set Minesweeper Consistency

The Minesweeper consistency problem that we investigated with restricted sets operated under a definition that the completed Minesweeper board is not allowed to have numbers outside the specified set. What if we relaxed this constraint such that numbers outside the specified set are allowed in the input board, but the completed board (which is to be found consistent with the input board) was allowed to have cells containing other numbers?

4.3 Other Minesweeper Problems

Minesweeper consistency is not the only interesting problem to analyze for complexity. The Minesweeper inference problem is the problem of determining whether there exists a yet-uncovered square in an input board whose contents can be deduced [5]. This problem in general turns out to be coNP-complete, where coNP is the complexity class of decision problems whose complement is in NP [2]. The problem that likely closest emulates the decisions of a Minesweeper player, locating all mines with highest probability, is PP-hard, where PP is the set of decision problems solvable in probabilistic polynomial time with error probability of less than $1/2$ in all instances [1]. An analysis of these two problems over possible ranges could be interesting.

4.4 Infinite Boards

We only analyzed range and complexity of variants of finite boards. But Minesweeper on an infinite board is a natural extension. There's several types of infinite Minesweeper boards that could be considered, for instance, boards with a finite height but extending infinitely in the width dimension, or boards extending infinitely in both dimensions, having no edges whatsoever. Several Minesweeper spinoff games implement this [6].

There exist ranges that are valid on infinite-in-both-dimensions Minesweeper boards but invalid on finite rectangular Minesweeper boards. The range $\{0, 3\}$ is proved invalid in finite rectangular boards via the 0-3 Gap Impossibility Theorem, but in an infinite board, one could construct an infinite straight wall of 3s, with only 0s on one side and only mines on the other side, producing this range.

Likewise, there exist ranges that are valid on finite rectangular Minesweeper boards but invalid on infinite-in-both-dimensions Minesweeper boards. The range $\{2, 7\}$ is proved valid in the individual cases above, but is invalid in the infinite board: consider a gap analysis on the closest 2-7 pair as was done in the impossibility proofs above, but note that now there must be eight cells around the 2 now due to the Minesweeper board being infinite.

Complexity questions regarding infinite boards could also be interesting. Consider, for instance, a finite description that details a pattern for specifying the contents of a not-necessarily-finite subset of the cells of an infinite board. What is the complexity of the Minesweeper consistency problem under restricted sets now?

4.5 Non-Rectangular Boards

Squares aren't the only regular polygons that tessellate. One could imagine Minesweeper being played on a triangular or hexagonal grid, and indeed implementations of such exist [7] [8] [9].

If we carry over the rule from conventional Minesweeper that a shared corner is all that is needed to constitute adjacency, triangular Minesweeper may be a more daunting task to analyze than rectangular Minesweeper, as each triangle has 12 adjacent triangles by this definition. If side-adjacency is required, triangular Minesweeper is likely much easier to analyze, with only 3 possible adjacencies. This distinction does not matter in hexagonal Minesweeper, as a tile does not have other tiles adjacent via only a corner.

4.6 Boards in More than Two Dimensions or Other Topologies

One could also consider Minesweeper played on a cube, and in combination with considerations of the above section, polyhedra other than a cube, like a tetrahedron. One nice variant is Minesweeper on a torus, where we identify opposite sides of a rectangular board, which would avoid the complex boundary effects we see with rectangular Minesweeper.

Bibliography

[1] de Bondt, Michiel. "The computational complexity of Minesweeper." arXiv:1204.4659 (2012).

[2] Hearn, Robert A., and Erik D. Demaine. *Games, Puzzles, and Computation.* CRC Press, 2009.

[3] Kaye, Richard. “Minesweeper is NP-complete." *The Mathematical Intelligencer* 22.2 (2000): 9-15.

[4] Mulzer, Wolfgang, and Günter Rote. “Minimum-weight triangulation is NP-hard." *Journal of the ACM* 55.2 (2008): 11.

[5] Scott, Allan, Ulrike Stege, and Iris van Rooij. “Minesweeper may not be NP-complete but it is hard nonetheless." *The Mathematical Intelligencer* 33:4 (2011), 5-17.

[6] JD Software LLC. “Minesweeper.io – Multiplayer Minesweeper Online," http://minesweeper.io/game.html.

[7] Minesweeperonline.net. “Hex Mines – Hexagonic Version of Minesweeper," http://www.minesweeperonline.net/hex-mines.php.

[8] Next Realm Games. “HexSweep – Multiplayer Online Hexagonal Minesweeper," http://hexsweep.io.

[9] WebMinesweeper. “WebMinesweeper – Play Free Online Minesweeper," http://webminesweeper.com.

[10] Wikipedia. “Minesweeper (video game)", https://en.wikipedia.org/wiki/Minesweeper_(video_game).

[11] Wikipedia. “Microsoft Minesweeper", https://en.wikipedia.org/wiki/Microsoft_Minesweeper.