Minesweeper

Knowledge Representation and Learning

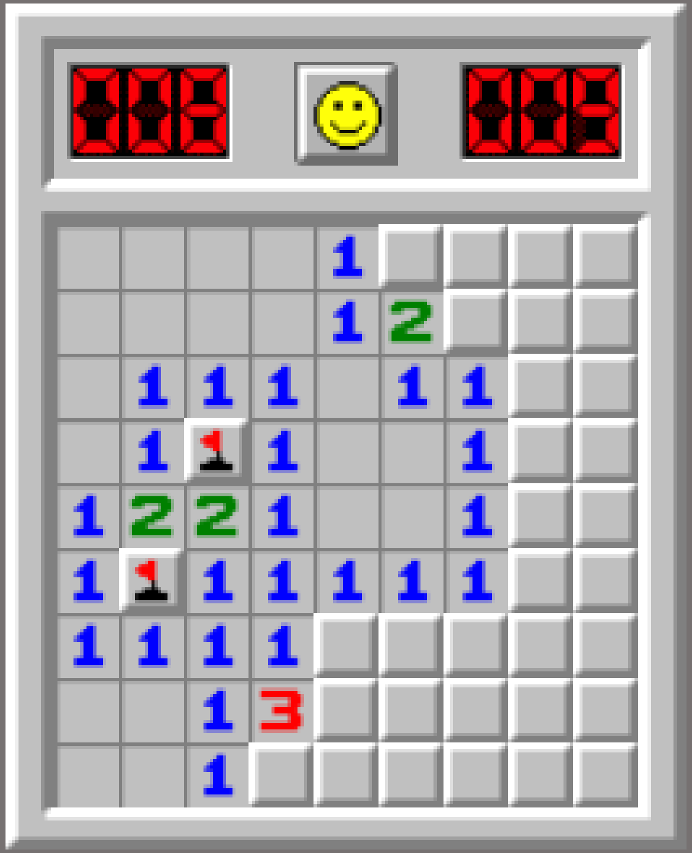
Akbota Norzhanova
(student id: 2106050)

Presentation rundown

1. Project description
2. Propositional logic
3. Project implementation
4. Demo

Project description

- **Minesweeper** is a game where mines are hidden in a grid of squares.
- **Game objective:** open all cells that do not contain mines
- If you open a cell with a mine, you lose. If you open a cell without a mine, it will show number of mines around that cell.
- **Project objective:** encode a game state in a state of formulas and use SAT to determine cells that contain/don’t contain a mine. If SAT cannot decide, use model counting to minimize probability of selecting a mine



Propositional formulization

Language

- For each cell $c_{i,j}$ $(i, j \in [0, \mathrm{m}))$, $Mine_{i,j}$ is propositional variable that represents the fact that the cell contains a mine
- Note: In the project, I assigned an index to each cell, such that variable x_1 meant cell with index 1 contains mine, and $\neg x_1$ means it is safe

Axioms

- Each cell can either contain a mine or not, but not both.
- There are N mines in total:

$$\bigwedge_{\substack{I\subseteq[m^2]\\|I|=m^2-N+1}} \bigvee_{i\in I} x_i \wedge \bigwedge_{\substack{I\subseteq[m^2]\\|I|=N+1}} \bigvee_{i\in I} \neg x_i$$

- If open safe cell c has n adjacent mines:

$$\bigwedge_{\substack{A\subseteq[adj(c)]\\|A|=adj(c)-n+1}} \bigvee_{a\in A} x_a \wedge \bigwedge_{\substack{A\subseteq[adj(c)]\\|A|=n+1}} \bigvee_{a\in A} \neg x_a$$

Game flow in the project

Game initialization

Player: defines number of mines N and number of cells per row m

Game: generates minesweeper field with given parameters and encodes “there are N mines in total”

1

First step

Player: opens the first cell

Game: if first opened cell is not mine, adds clauses on number of mines around the opened cell.

2

Continue playing

Game: provides suggestions based on encoded clauses (and opens/flags them).

Game: continues playing until all safe cells are open and no mines are opened during the game

3

Project models

Mode 1

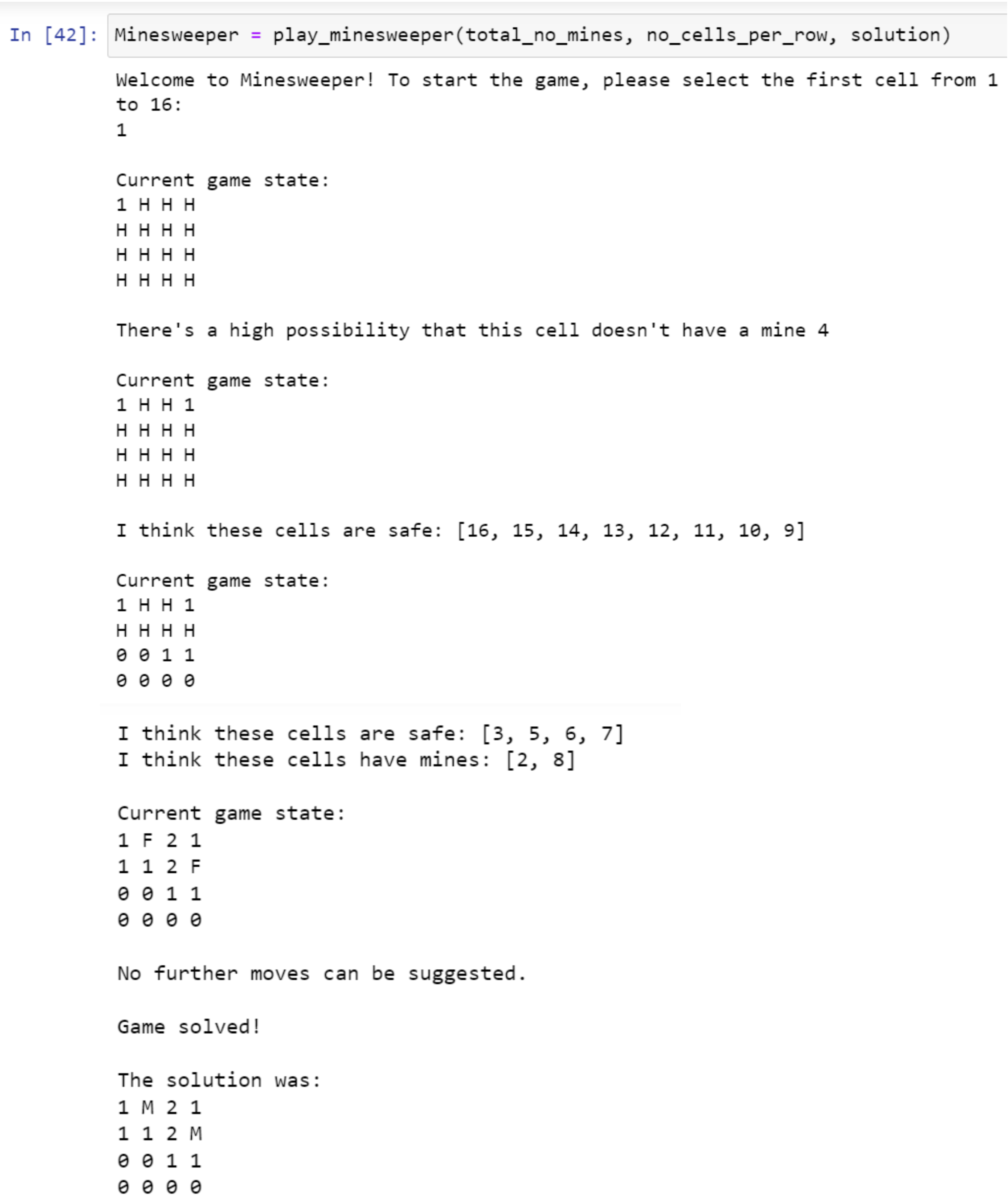
jupyter
SAT

Mode 2

jupyter
SAT
pygame

Mode 1

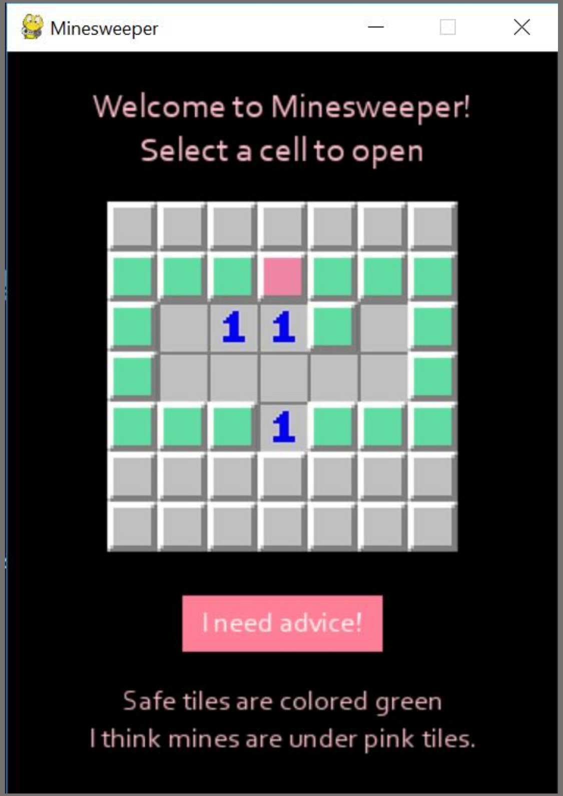
- Game is encoded as a function.
- Player selects the first cell to open and pysat decides what cells to open. If no exact solution can be suggected, pysat performs model counting.
- When pysat suggests a mine, it is flagged as ‘F’.



```
In [42]: Minesweeper = play_minesweeper(total_no_mines, no_cells_per_row, solution)

         Welcome to Minesweeper! To start the game, please select the first cell from 1
         to 16:
         1

         Current game state:
         1 H H H
         H H H H
         H H H H
         H H H H

         There's a high possibility that this cell doesn't have a mine 4

         Current game state:
         1 H H 1
         H H H H
         H H H H
         H H H H

         I think these cells are safe: [16, 15, 14, 13, 12, 11, 10, 9]

         Current game state:
         1 H H 1
         H H H H
         0 0 1 1
         0 0 0 0

         I think these cells are safe: [3, 5, 6, 7]
         I think these cells have mines: [2, 8]

         Current game state:
         1 F 2 1
         1 1 2 F
         0 0 1 1
         0 0 0 0

         No further moves can be suggested.

         Game solved!

         The solution was:
         1 M 2 1
         1 1 2 M
         0 0 1 1
         0 0 0 0
```

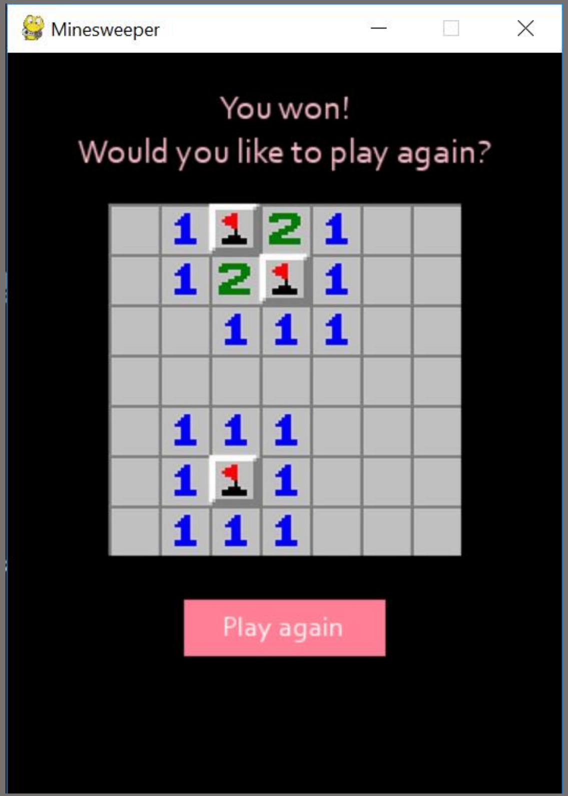
Mode 2

- Game is encoded as pygame game.
- Player opens cells one by one, but can ask for a hint from the game anytime.
- If hint is asked, safe cells are colored green and cells with mines - red. If no exact solution can be suggested, cells with the highest probability to be safe are colored blue.



Welcome to Minesweeper!
Select a cell to open

Safe tiles are colored green
I think mines are under pink tiles.



You won!
Would you like to play again?

Demo