CS228 Assignment 1

Dhanush Sanjay Nidamanuri-24B0968

Siddharthi Vivekanand-24B1056

August 2025

Contents

Chapter 1

SAT-Based Sudoku Solver

1.1 Encoding the variables

Each ordered triplet $(i, j, n) \in [0, 8] \times [0, 8] \times [0, 8]$ is mapped to a unique postive integer by the function `var()`. This results in a *bijection*(for a selected co-domain). This mapped integer represents the index of the corresponding propositional variable.

$$var(i, j, n) = 100(n + 1) \; + \; 10(i + 1) \; + \; (j + 1)$$

Where,

- i = Index of the row.
- j = Index of the column.
- n = Number in 0-8.

Explanation:

1. For each n there are 100 places available but only 81 are being utilised. These places define the *variable space* for a unique n.
2. The factor n scales this space up by 100 so that no variables with different n can map to the same value in the *co-domain*.
3. And the offset $10(i + 1) + (j + 1)$ encodes each cell uniquely.

So, each triplet (i, j, n) correspond to a unique positive integer variable index.

1.2 Implementing Constraints

- Any potential sudoku solution must satify 4 conditions :
  1. Each **row** must contain every number 1-9.
  2. Each **column** must contain every number 1-9.
  3. Each 3×3 **subgrid** ranging from rows $3i + 1$ to $3i + 3$ and columns $3j + 1$ to $3j + 3$, where $i, j \in \{0, 1, 2\}$ must contain every number 1-9.
  4. Each **cell** must not contain more than one number.
- **Rows:**

$$\bigwedge_{i=0}^{8} \bigwedge_{n=0}^{8} \bigvee_{j=0}^{8} var(i, j, n)$$

- **Columns:**

$$\bigwedge_{j=0}^{8} \bigwedge_{n=0}^{8} \bigvee_{i=0}^{8} var(i, j, n)$$

- **Subgrids:**

$$\bigwedge_{i\in\{0,3,6\}} \bigwedge_{j\in\{0,3,6\}} \bigwedge_{n=0}^{8} \left(\bigvee_{i_1=0}^{2} \bigvee_{j_1=0}^{2} var(\,i+i_1,\,j+j_1,\,n \right)$$

- **Cells:**

$$\bigwedge_{i=0}^{8} \bigwedge_{j=0}^{8} \bigwedge_{\substack{n_1,n_2=0\\ n_1\neq n_2}}^{8} (\neg\, var(i,j,n_1) \,\vee\, \neg\, var(i,j,n_2))$$

- **Filled cells:**

$$\bigwedge_{i=0}^{8} \bigwedge_{j=0}^{8} \bigwedge_{n=0}^{8} if(n\ in\ cell)\ var(i,j,n)$$

- Since each of these constraints must be satisfied together, the final formula is the conjunction of all of the mentioned conditions.
- Thus, each condition is encoded as a set of cnf clauses.
- Finally, all the clauses are fed into the solver. If the solver fetches a satisfying assignment, we decode the assignment into a resultant grid and return it.

1.3 Decoding the model

After fetching the model, we iterate through the truth values for each variable and if any variable is set to true, we decode that variable's index as:

$$i = \left\lfloor \frac{x \bmod 100}{10} \right\rfloor - 1,\ j = (x \bmod 10) - 1,\ n = \left\lfloor \frac{x}{100} \right\rfloor$$

- $x =$ The index of the propositional variable.

Since the function `var()` was a bijection, the inverse of it exists and is also a bijection. So, decoding 2 different variables indices always yields different triplets of (i, j, k). We assign the number n of that variable to the corresponding cell (i, j) of the resultant grid. Thus, the resultant grid is obtained.

1.4 Contributions

We both have solved this question in parallel as it doesnt take much time. The final code we included in this have parts from both of our codes , we selected some parts from each other codes(based on optimality and speed) and combined them.

Chapter 2

SAT-Based Sokoban Solver

2.1 Variable Encoding Logic

We used padding of walls on all four sides of the board in order to avoid corner case logic.The main idea behind this encoding is to divide the board into T+1 timestamps from 0 to T and each entity's position is calculated by the formula. Since we have added padding, The following changes were needed:

- Number of rows and columns of new Puzzle will be $N + 2$ and $M + 2$ respectively where N and M stand for number of rows and columns respectively in the original puzzle.
- So, During parsing we added a offset 2 to each coordinates of the initial conditions so that all those coordinates become positive (required for variable encoding since SAT solver expects positive variables).

2.1.1 Player Position Encoding

We assign a unique variable ID to represent the player being at cell (x, y) at time step t using the formula:

$$\text{var\_player}(x, y, t) = (N + 3)(M + 2)t(1 + \text{num\_boxes}) + (M + 2)x + y \tag{2.1}$$

where

- N = Number of rows in the grid
- M = Number of columns in the grid
- t = Time stamp
- num_boxes = Number of boxes in the puzzle
- x, y = Grid coordinates of the player

Explanation:

1. The term $(N + 3)(M + 2)(1 + \text{num\_boxes})$ defines the *variable space* for a unique time.
2. The factor t scales this *variable space*, ensuring that variables from different time do not overlap.
3. The local offset $(M + 2)x + y$ uniquely encodes the player's position within the grid at that time.

Thus, each triple (x, y, t) corresponds to a distinct positive integer variable ID suitable for SAT encoding.

2.1.2 Box Position Encoding

We assign a unique variable ID to represent the each Box being at cell (x, y) at time step t using the formula:

$$\begin{aligned}\text{var\_box}(b, x, y, t) = {} & (N + 3)(M + 2)t(1 + \text{num\_boxes}) \\ & + (N + 3)(M + 2)b + (M + 2)x + y\end{aligned} \tag{2.2}$$

where

- N = Number of rows in the grid
- M = Number of columns in the grid
- t = Time stamp
- num_boxes = Number of Boxes in the puzzle
- x, y = Grid coordinates of the BOx
- b = Box id of ith box

Explanation:

1. The term $(N+3)(M+2)(1+\text{num\_boxes})$ defines the *variable space* for a unique time.
2. The factor t scales this *variable space*, ensuring that variables from different time do not overlap.
3. The local offset $(M+2)x+y$ uniquely encodes the player's position within the grid at that time.

Thus, each triple (b, x, y, t) corresponds to a distinct positive integer variable ID suitable for SAT encoding.

2.2 Game Logic to CNF Conversion

The Constraints we encoded in broad sense are:

1. Initial conditions of players and Boxes
2. Player cannot be in Padding or Walls at any time
3. Player can only move to left or right or up or down
4. Player cannot be in two places simultaneously at any Time
5. A box may be pushed in any of the four directions if, The box is in the cell directly adjacent to the player in the chosen direction,and the box is pushed to The cell immediately beyond the box, in the same direction.
6. Box cannot be in padding or walls at any time
7. Box cannot be in two places Simultaneously
8. Player cannot overlap Boxes at any time
9. Two Distinct boxes cannot overlap at any time
10. At time T, all boxes should in a goal

2.2.1 Initial conditions of players and Boxes

At time $T = 0$, the player and all boxes are fixed to their starting positions.Thus, the initial configuration is represented by adding unit clauses to our CNF for the player and each box at $T = 0$.It is encoded using:

- **Player condition:**
$$P(\text{player}, \text{player\_start}_x, \text{player\_start}_y, 0)$$
- **Boxes condition:**
$$\bigwedge_{b \in boxes} B(\text{b}, \text{b\_start}_x, \text{b\_start}_y, 0)$$

2.2.2 Player Movement

Player cannot be in padding or walls at time.It is encoded using:

$$\bigwedge_{t=0}^{T}\left(\bigwedge_{i=1}^{N+2} \neg P(i,1,t) \wedge \neg P(i,M+2,t)\right) \wedge \left(\bigwedge_{j=1}^{M+2} \neg P(1,j,t) \wedge \neg P(N+2,j,t)\right) \wedge \left(\bigwedge_{w\,\in\,walls} \neg P(w_x,w_y,t)\right) \tag{2.3}$$

And if player is at (x, y) at time $t + 1$,then the conditions for players position at time t is encoded in these:

$$\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} \neg P(i,j,t+1) \vee P(i,j,t) \vee P(i-1,j,t) \vee P(i+1,j,t) \vee P(i,j-1,t) \vee P(i,j+1,t) \tag{2.4}$$

And if player is at (x, y) at time t ,then the conditions for players position at time $t + 1$ is encoded in these:

$$\begin{gathered}\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} \neg P(i,j,t) \vee P(i,j,t+1) \vee P(i-1,j,t+1)\\ \vee P(i+1,j,t+1) \vee P(i,j-1,t+1) \vee P(i,j+1,t+1)\end{gathered} \tag{2.5}$$

The Clauses which ensure that player is not in two places simultaneously are encoded in these:

$$\bigwedge_{t=0}^{T}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1}\bigwedge_{i_1=2}^{N+1}\bigwedge_{j_1=2}^{M+1} if(i_1 \neq i \,||\, j_1 \neq j) \;\; (\neg P(i,j,t) \vee \neg P(i_1,j_1,t))$$

2.2.3 Box Movement

if a box b is at (x, y) at time T then, it can either be at (x, y) or $(x + 1, y)$ or $(x - 1, y)$ or $(x, y + 1)$ or $(x, y - 1)$,it is encoded in:

$$\begin{gathered}\bigwedge_{b\,\in\,boxes}\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} \neg B(b,i,j,t+1) \vee B(b,i,j,t) \vee B(b,i-1,j,t)\\ \vee B(b,i+1,j,t) \vee B(b,i,j-1,t) \vee B(b,i,j+1,t)\end{gathered} \tag{2.6}$$

if a box b is at (x, y) at time $T + 1$, then the box can be at,

- $(x - 1, y)$ at T and the corresponding player constraints are encoded in:

$$\begin{gathered}\bigwedge_{b\,\in\,boxes}\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} (\neg B(b,i-1,j,t) \vee \neg B(b,i,j,t+1) \vee P(i-2,j,t))\\ \wedge (\neg B(b,i-1,j,t) \vee \neg B(b,i,j,t+1) \vee P(i-1,j,t+1))\end{gathered} \tag{2.7}$$

- $(x + 1, y)$ at T and the corresponding player constraints are encoded in:

$$\begin{gathered}\bigwedge_{b\,\in\,boxes}\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} (\neg B(b,i+1,j,t) \vee \neg B(b,i,j,t+1) \vee P(i+2,j,t))\\ \wedge (\neg B(b,i+1,j,t) \vee \neg B(b,i,j,t+1) \vee P(i+1,j,t+1))\end{gathered} \tag{2.8}$$

- $(x, y - 1)$ at T and the corresponding player constraints are encoded in:

$$\begin{gathered}\bigwedge_{b\,\in\,boxes}\bigwedge_{t=0}^{T-1}\bigwedge_{i=2}^{N+1}\bigwedge_{j=2}^{M+1} (\neg B(b,i,j-1,t) \vee \neg B(b,i,j,t+1) \vee P(i,j-2,t))\\ \wedge (\neg B(b,i,j-1,t) \vee \neg B(b,i,j,t+1) \vee P(i,j-1,t+1))\end{gathered} \tag{2.9}$$

- $(x, y+1)$ at T and the corresponding player constraints are encoded in:

$$\bigwedge_{b\ \in boxes} \bigwedge_{t=0}^{T-1} \bigwedge_{i=2}^{N+1} \bigwedge_{j=2}^{M+1} (\neg B(b,i,j+1,t) \vee \neg B(b,i,j,t+1) \vee P(i,j+2,t)) \\ \wedge (\neg B(b,i,j+1,t) \vee \neg B(b,i,j,t+1) \vee P(i,j+1,t+1)) \tag{2.10}$$

The conditions for a box b to be in two places simultaneously at any time T is encoded as:

$$\bigwedge_{b\ \in boxes} \bigwedge_{t=0}^{T} \bigwedge_{i=2}^{N+1} \bigwedge_{j=2}^{M+1} \bigwedge_{i_1=2}^{N+1} \\ \bigwedge_{j_1=2}^{M+1} if(i_1 \neq i \,||\, j_1 \neq j) \;\; (\neg B(b,i,j,t) \vee \neg B(b,i_1,j_1,t)) \tag{2.11}$$

The conditions for a box b to be not in padding or walls at any time T is encoded as:

$$\bigwedge_{b\ \in boxes} \bigwedge_{t=0}^{T} \left(\bigwedge_{i=1}^{N+2} \neg B(b,i,1,t) \wedge \neg B(b,i,M+2,t) \right) \\ \wedge \left(\bigwedge_{j=1}^{M+2} \neg B(b,1,j,t) \wedge \neg B(b,N+2,j,t) \right) \wedge \left(\bigwedge_{w\ \in walls} \neg B(b,w_x,w_y,t) \right) \tag{2.12}$$

2.2.4 Non-overlap constraints

The conditions for the player and a box b to be not in same place simultaneously at any time T and The condition for two distinct boxes $b, b1$ to be not in same place simultaneously at any time T is encoded in:

$$\bigwedge_{t=0}^{T} \bigwedge_{i=2}^{N+1} \bigwedge_{j=2}^{M+1} \bigwedge_{b_1\ \in boxes} (\neg P(i,j,t) \vee \neg B(b_1,i,j,t)) \wedge \bigwedge_{\substack{b_2 \in boxes \\ b_2 \neq b_1}} (\neg B(b_1,i,j,t) \vee \neg B(b_2,i,j,t)\ ) \tag{2.13}$$

2.2.5 Goal Conditions

The conditions for it to be SAT are that all boxes should be in a distinct goals by time T.It is encoded as:

$$\bigwedge_{b \in boxes} \bigvee_{g \in goals} B(b, g_x, g_y, T)$$

2.3 Decoding the SAT Solvers Output

Since the output of solver will be the variables which were encoded previously we need to inverse the previous bijection to get the states of boxes,players at each time stamp.But we only need to output the player positions at each time t from 0 to T. From encoding scheme we know that player position lies range at any time t. at any time t from 0 to T:

$$(N+3)(M+2)t(1+\text{num\_boxes}) \ \leq \ P(i,j,t)$$

$$P(i,j,t) \ \leq \ (N+3)(M+2)t(1+\text{num\_boxes}) + (N+3)(M+2)$$

therefore we filter the variables from SAT model which lie in this range and the decode them using the function 2.14 and store them in a map *coords* of type $< int, < int, int >>$ where the first *int* is the time and the pair of *int* are coordinates at that time.

$$\text{coords}\Big(\Big\lfloor \frac{v}{a} \Big\rfloor\Big) = \Big[\Big\lfloor \frac{v \bmod a}{M+2} \Big\rfloor ,\ v \bmod (M+2) \Big] \tag{2.14}$$

Where:

- $a = (N+3)(M+2)(1+B)$
- $v =$ a SAT variable from the model

Now the steps taken between time t an d $t+1$ are calculated using dx, dy for $t, t+1$ and then they are mapped correspondingly to the four directions *UP,DOWN,LEFT,RIGHT* and added to to list *moves* and then returned.

2.4 Contributions

- Both of us have done the Logic Designing and constraints identification part collaboratively.
- Both of us have contributed to design the encoding function
- Both of us have contributed to design the decoding function
- Dhanush Sanjay have written the part of code, which contains coding and adding all constraints to CNF and also the part which contains encoding the variables to write CNFs.
- Vivekanand have written the part of which decodes positions of player from SAT solvers model.
- Both of us have taken part in writing latex report for this question
- Both of us took part in Debugging in the Code.