Final Project - Formal Verification

**Part 1**

1. Define an FDS for a general n × m Sokoban board. Use XSB format to describe the board.

Warehouse keeper is on the floor and next to him is a floor:

Warehouse keeper is on the floor and next to him is a wall:

Warehouse keeper is on the goal and next to him is a wall:

Warehouse keeper is on the floor and next to him is a goal:

Warehouse keeper is on the goal and next to him is a goal:

Warehouse keeper is on the goal and next to him is a floor:

Warehouse keeper is on the floor and next to him is a box next to a floor:

Warehouse keeper is on the goal and next to him is a box next to a floor:

Warehouse keeper is on the floor and next to him is a box next to a wall:

Warehouse keeper is on the goal and next to him is a box next to a wall:

Warehouse keeper is on the floor and next to him is a box next to a box:

Warehouse keeper is on the floor and next to him is a box next to a goal:

Warehouse keeper is on the goal and next to him is a box next to a goal:

Warehouse keeper is on the goal and next to him is a box next to a box:

Warehouse keeper is on the floor and next to him is a box on goal next to a floor:

Warehouse keeper is on the goal and next to him is a box on goal next to a floor:

Warehouse keeper is on the floor and next to him is a box on goal next to a wall:

Warehouse keeper is on the goal and next to him is a box on goal next to a wall:

Warehouse keeper is on the floor and next to him is a box on goal next to a box:

Warehouse keeper is on the goal and next to him is a box on goal next to a box:

Warehouse keeper is on the floor and next to him is a box on goal next to a box on goal:

Warehouse keeper is on the goal and next to him is a box on goal next to a box on goal:

Warehouse keeper is on the floor and next to him is a box next to a box on goal:

Warehouse keeper is on the goal and next to him is a box next to a box on goal:

-A Boolean value that holds true if it is possible to move in that direction

𝐷 = {𝑉, , , 𝐽, 𝐶}

𝑉 = {B , , , , , , }

𝐽 = ??

𝐶 =??

B - a general n × m Sokoban board in XSB format.

position of the warehouse keeper on board -

2. Define a general temporal logic specification for a win of the Sokoban board.

Winning condition: a temporal logic expression that specifies when the game is won. This involves checking if all boxes are on goal cells.

winning\_condition = (box1\_on\_goal && box2\_on\_goal && ... && boxN\_on\_goal) or in different words "there are no $ on the board" (there are no boxes on the board that are not on goal)

Express in Temporal Logic:

Winning = finally ($)

# for running nuXmv we used the following commands:

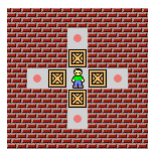
nuXmv -int

read\_model -i sokoban.smv (smv file name)

go

check\_ltlspec

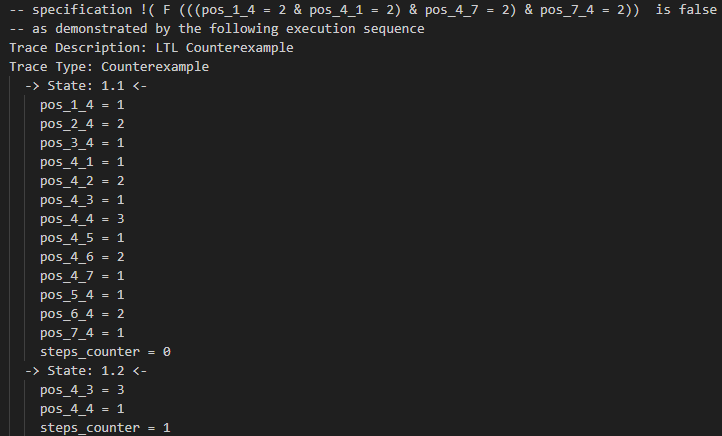
**Board 1:**

****

Part 2:

תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי



תמונה שמכילה טקסט, גופן, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטי

Part 4:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**Board 2:**

Part 2:

תמונה שמכילה מלבן, סרט מצויר, פיקסל

התיאור נוצר באופן אוטומטי

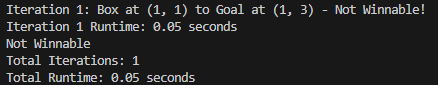
תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי



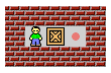


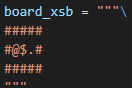
Part 4:



**Board 3:**

Part 2:







תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

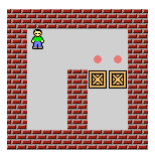
Part 4:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

**Board 4:**

Part 2:

****

**תמונה שמכילה צילום מסך, טקסט, מספר, קו

התיאור נוצר באופן אוטומטי**

****



Part 4:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**Part 3:**

The choice between BDD and SAT-based model checking depends on the specific characteristics of the problem at hand. BDDs are advantageous for problems with a high degree of regularity and require symbolic manipulation, but they suffer from memory limitations. SAT solvers, on the other hand, excel in handling large, sparse, and complex state spaces efficiently, benefiting from modern advancements in solver technology. For many modern verification tasks, SAT-based approaches are preferred due to their scalability and performance.

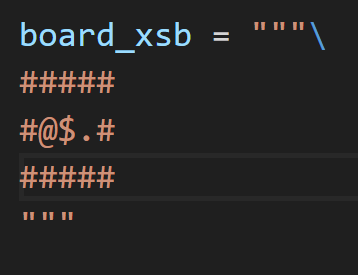
Not winnable board, SAT is more efficient:

**תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי**

Winnable board, BDD is more efficient:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

Winnable board, SAT is more efficient:

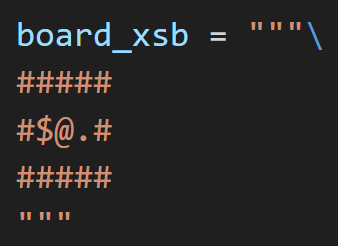
תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

Not winnable board, BDD is more efficient:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

As we can see, for structural boards the bdd engine is more efficient and for more complicated boards the sat engine is more efficient.