1. Define the agent design problem

An agent refers to a computer program that’ll act autonomously to achieve the desired goal by gathering and deducing the information based on environment. The problem when there’s need to create a specific agent to achieve particular output is what we called agent design problem. The agent should be able to learn from the provided data and act rationally to achieve the goal.

1. What type of agent is it?

-Discrete/ Continuous

As the states are always discrete in this situation, we could say it’s discrete

-Deterministic / Non-deterministic

After performing an action (moves the player) from a state, the next stage is determined and agent knows exactly what stage it’ll be in. So it’s deterministic

-Fully observable/ Partially

The percept function in this case is one-to-one mapping so the agent always know exactly which stage it’ll be in. Hence it’s fully observable

-Static/ dynamic

If a move is not performed (when agent is calculating for the next action), the stage will always stay the same so we can say it’s static

For game Sokoban, there’re several common deadlocks which can be systematically summed up. Below is the description on the most frequently seen deadlocks and how to implement them in the code.

1. Dead square deadlocks

This type of deadlock will immediately create a unsolvable situation when pushing a box to specific space. The most common seen one is when a box is pushed near the wall but there’re not targets on that row. And similar, when a box is in the corner (with 2 directions facing wall) but not already on a target, it becomes unsolvable. To implement the function in code, algorithm is shown as below.

For BOX in ALL\_BOXS {

If BOX near the WALL{

If not Target on the row {Return deadlock = True}

If Targets on the row{

If all Targets been taken { Return deadlock = True }

}

If BOX in the corner & not on Target { Return deadlock = True }

}

1. Freeze deadlocks

Besides being pushed near the wall, there’re other movements might cause the deadlocks In freeze deadlocks, all the boxes become immoveable. Player is unable to move when she/he’s facing a box and try to move towards that direction. For example, if a box is on Player’s right side and player wants to perform an action moving right but the action in

To avoid this situation, we use algorithm shown as below.