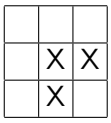
1. This question considers a variant of Tic-Tac-Toe, called Notakto, in which both players alternately place a cross (no one uses circles) on a 3-by-3 board, and the player that produces 3 crosses in a row (horizontal, vertical or diagonal) *loses* the game.

   After 3 moves, the following game state s_0 is given, with player $p(s_0) = -1$ making the next move:

   [You may want to draw the decision tree starting at this state. Considering transpositions and symmetries can make this much easier!]

   (a) (3 points) What is the limit of the random rollout value $\lim_{N(s_0)\to\infty} \tilde{V}_{\mathrm{R}}(s_0)$?

   (b) (2 points) Next we implement a smart rollout heuristic: whenever possible, we select actions that do not end the game. What is the limit of the random rollout value using this heuristic?

   (c) (2 points) What is the minmax value $V(s_0)$? What is the effect of the above heuristic rollout on the minmax value? Explain your answer.

   (d) (3 points) Override the following standard implementation of `rollout()` to implement the smart rollouts of the above sub-question (b). Make sure you implement the above heuristic, and do not use an abstract heuristic $H(s, a)$.

```
def rollout(self, node):
    state = node.state
    while not state.terminal():
        action = random.choice(state.actions())
        state = state.transition(action)
    return state.reward()
```

 End of exam