Problem

The Japanese game *go-moku* is played by two players, “X” and “O,” on a 19 × 19 grid. Players take turns placing markers, and the first player to achieve five of her markers consecutively in a row, column, or diagonal is the winner. Consider this game generalized to an n × n board. Let

$$GM = \{\langle B\rangle \mid B \text{ is a position in generalized go-moku, where player “X” has a winning strategy}\}.$$

By a *position* we mean a board with markers placed on it, such as may occur in the middle of a play of the game, together with an indication of which player moves next. Show that *GM* ? PSPACE.

Step-by-step solution

Step 1 of 3

PSPACE: PSPACE is deterministic Turing machine that contains the class of languages that are decidable in polynomial space on a deterministic Turing machine *i.e.,* $\mathrm{PSPACE} = \bigcup_k SPACE\left(n^k\right)$

To generalize *go – moku* game on $n \times n$ board is given as

$GM = \{\langle B\rangle \mid B \text{is a position in generalized } go\text{ - }moku\text{, where player "X" has a winning strategy}\}$

- This game is play by 2 players "X" and "O" Now we have to prove that $GM \in PSPACE$.
- Let us assume that B is written as a grid of "X" and "O" are empty, so the length of the input is $O\left(n^2\right)$
- Now let us define a recursive algorithm to solve $GM(B)$, which accepts if there is a winning strategy for player X starting at position B.

Comment

Step 2 of 3

Recursive Algorithm of *GM* :

$GM(B)$:

(1) potential X moves: All spaces of i in position B without marker on them

(i) Put an X marker on space i, next changing the position to B'. If there are $5X's$ in a row (it a best move) then accept. If the board is now full, and no one has won, reject.

(ii) potential O moves: all spaces j in position B' without _ markers on them.

(a) Put an O marker on space j, next changing the position to B'. if there are $5\,O's$ in a row(it is also best move) or the board in full and no one has won, loop to the next *i* (go to step(i)); putting an X on i is obviously a bad move.

(b) Otherwise, run $\mathrm{GM}(\mathrm{B}'')$

- If it accepts, loop to next j (goto step (b)).
- If it rejects, loop to the next i (goto step (i))

(iii) If all j cause $\mathrm{GM}(\mathrm{B}'')$ to accept, i is a good X move, since it covers all possible O moves, so accept.

Comment

Step 3 of 3

(2) If no *i* step (i) causes accept, reject, there are no good moves from this position, so reject.

- We can loop through all moves i, j at each step, since we can just reuse the space.
- As w just need to store configurations B', B'', which takes only $O(n^2)$ space.
- And our recursion is only $O(n^2)$ since there are at most n^2 moves.
- Total space needed is $O(n^4)$, which is a polynomial in the input length, since we assumed the input had length $O(n^2)$.
- Clearly no game of generalized *go – moku* on $n \times n$ board can have more than $n \times n$ moves.
- So possible configurations following from *B* needs only polynomial spac*e.*
- Thus $GM \in PSPACE$

Comment