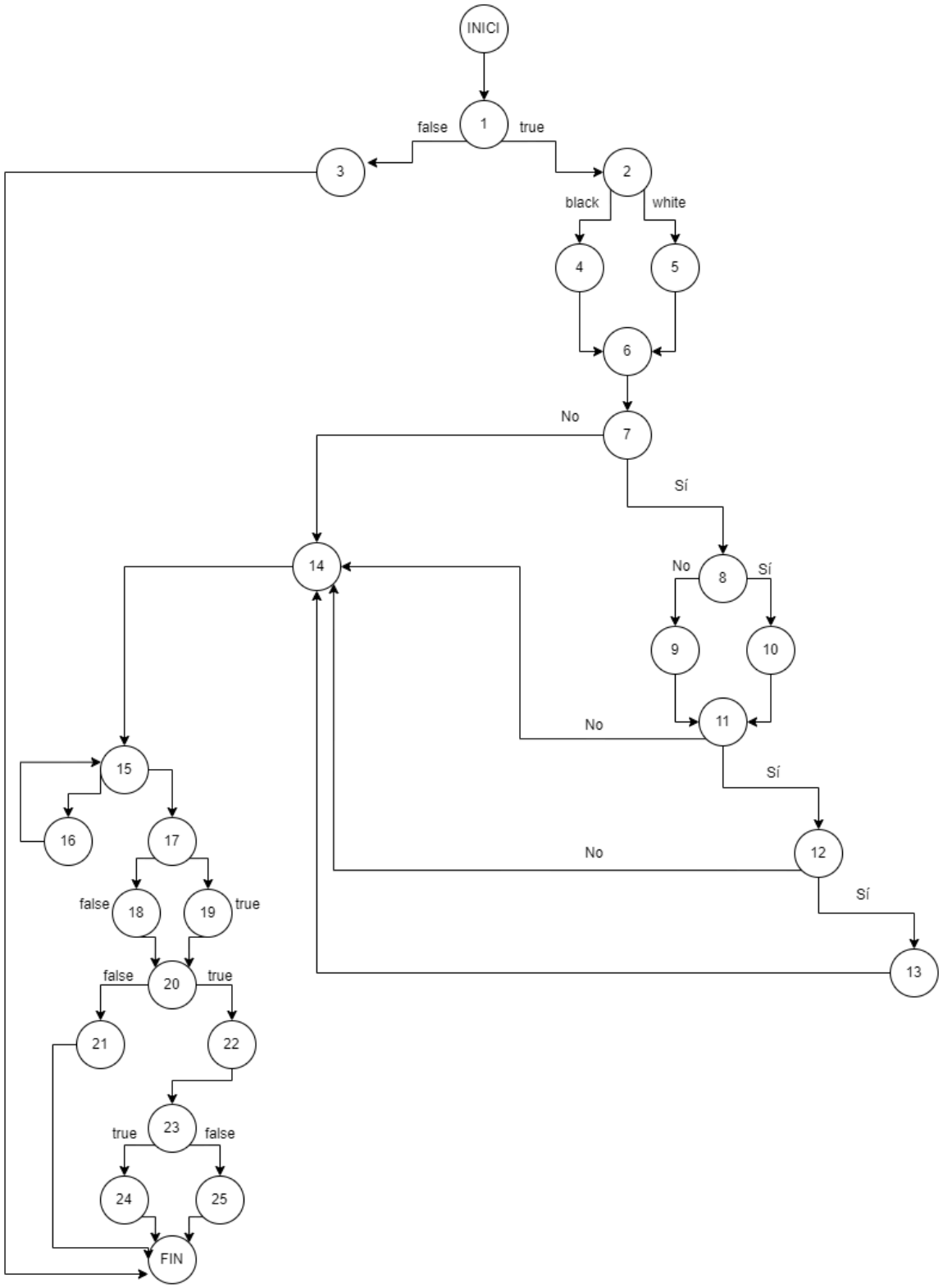
GRAFO CHESSMATE



COMPLEJIDAD CICLOMÁTICA

Número de regiones del grafo	11
Número de regiones cerradas	10 + 1
Aristas - Nodos + 2	11
Nodos predicado + 1	10 + 1

POSIBLES CAMINOS

C1	1-3-F
C2	1-2-(4 o 5)-6-7-14-15-16-15-17-18-20-21-F
C3	1-2-(4 o 5)-6-7-11-14-15-16-15-17-18-20-21-F
C4	1-2-(4 o 5)-6-7-11-12-14-15-16-15-17-18-20-21-F
C5	1-2-(4 o 5)-6-7-11-12-13-14-15-16-15-17-18-20-21-F
C6	1-2-(4 o 5)-6-7-14-15-16-15-17-19-22-23-(24 o 25)-F
C7	1-2-(4 o 5)-6-7-11-14-15-16-15-17-19-22-23-(24 o 25)-F
C8	1-2-(4 o 5)-6-7-11-12-14-15-16-15-17-19-22-23-(24 o 25)-F
C9	1-2-(4 o 5)-6-7-11-12-13-14-15-16-15-17-19-22-23-(24 o 25)-F

OBTENCIÓN DE LOS CASOS DE PRUEBA

Camino	Caso de prueba	Resultado esperado
1	bthinking es true	false
2	bthinking es false,p.enPassantSquare < 0 **o** p.board[move.from] != (player?ChessPosition.PAWN:-ChessPosition.PAWN), cumple el for(int i = 0; i < nMoves && !bOK ; i++) , no cumple el if(move.to == piece_moves[i]), entra en el if (!bOK) y return false;	false
3	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), no cumple este if (offset == 11 || offset == 9), cumple el for(int i = 0; i < nMoves && !bOK ; i++) , no cumple el if(move.to == piece_moves[i]), entra en el if (!bOK) y return false;	false

4	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), cumple este if (offset == 11 || offset == 9),no cumple if (p.board[move.from - 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN) **o** p.board[move.from + 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN)) cumple el for(int i = 0; i < nMoves && !bOK ; i++) , no cumple el if(move.to == piece_moves[i]), entra en el if (!bOK) y return false; piece_moves[i]), entra en el if (!bOK) y return false;	false
5	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), cumple este if (offset == 11 || offset == 9),cumple if (p.board[move.from - 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN) **o** p.board[move.from + 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN)),b0k = true cumple el for(int i = 0; i < nMoves && !bOK ; i++) , no cumple el if(move.to == piece_moves[i]), entra en el if (!bOK) y return false; piece_moves[i]), entra en el if (!bOK) y return false;	false
6	bthinking es false,p.enPassantSquare < 0 **o** p.board[move.from] != (player?ChessPosition.PAWN:-ChessPosition.PAWN), cumple el for(int i = 0; i < nMoves && !bOK ; i++) ,cumple el if(move.to == piece_moves[i]), no entra en el if (!bOK),dependiendo de si player es true o false, tendrá una salida u otra,if (player ? np.bWhiteChecked : np.bBlackChecked)	true o false, dependiendo del último if
7	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), no cumple este if (offset == 11 || offset == 9), cumple el for(int i = 0; i < nMoves && !bOK ; i++) ,cumple el if(move.to == piece_moves[i]), no entra en el if (!bOK),dependiendo de si player es true o false, tendrá una salida u otra,if (player ? np.bWhiteChecked : np.bBlackChecked)	true o false, dependiendo del último if
8	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), cumple este if (offset == 11 || offset == 9),no cumple if (p.board[move.from - 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN) **o** p.board[move.from + 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN)), cumple el for(int i = 0; i < nMoves && !bOK ; i++) ,cumple el if(move.to == piece_moves[i]), no entra en el if (!bOK),dependiendo de si player es true o false, tendrá una salida u otra,if (player ? np.bWhiteChecked : np.bBlackChecked)	true o false, dependiendo del último if
9	bthinking es false,p.enPassantSquare > 0 **y** p.board[move.from] == (player?ChessPosition.PAWN:-ChessPosition.PAWN),Entra o no en el if (offset < 0), cumple este if (offset == 11 || offset == 9),cumple if (p.board[move.from - 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN) **o** p.board[move.from + 1] == (player?-ChessPosition.PAWN:ChessPosition.PAWN)),b0k = true, cumple el for(int i = 0; i < nMoves && !bOK ; i++) ,cumple el if(move.to == piece_moves[i]), no entra en el if (!bOK),dependiendo de si player es true o false, tendrá una salida u otra,if (player ? np.bWhiteChecked : np.bBlackChecked)	true o false, dependiendo del último if