

READ ME TEMA 2 AC

Am ales 14 stari dupa cum urmeaza: 4 stari pentru directii(stanga, dreapta, sus, jos), 4 stari pentru verificarea directiilor, 4 stari pentru mutarea catre directia respectiva, o stare pentru directie, o stare pentru pozitia initiala.

Verificarea directiilor se face dupa mutare si anume: initial ma mut catre partea dreapta pentru ca asa am considerat de cuviinta din intelegerea algoritmului. In cazul in care, dupa mutare, verificarea da rezultatul 1(perete), ma mut inapoi in pozitia precedenta. In cazul in care este 0, maze_we ia valoarea 1(pentru a trece 2 pe unde am trecut). Initial, initializez maze_we cu 0(pentru ca inca nu am parcurs nimic si nu am unde sa pun 2, urmand a da alta valoare in momentul in care gasesc o casuta care se potriveste conditiilor).

Declar 2 registrii, anume:stare si stare urmatoare(pe acelasi numar de biti, anume 4). Folosesc 2 blocuri always, anume: unul pentru frontul crescator al clkului unde declar faptul ca starea in care ma aflu nu are cum sa fie mai mare decat starea urmatoare. Al doilea bloc always se modifica in momentul in care orice input se modifica(fiind logic, pentru ca se muta pozitia initiala, anume row si col). Declar si un registru deplasare cu care voi face deplasarea in ultima parte a programului.

Algoritmul l-am gandit in felul urmator: initial, ma mut spre dreapta cu o casuta, pe care o verific ulterior daca este sau nu valabila. In cazul in care in casuta respectiva exista 1(perete), ma intorc de unde am venit si schimb directia de deplasare. In cazul in care in casuta respectiva exista 0(gol), activez maze_we care pune 2 pe unde am fost deja. Am scris codul pentru mutari si pentru verificarile tuturor directiilor, respectand conditiile pe care le-am mentionat mai sus, existand cate o stare de verificare pentru fiecare directie, respectiv mutarea catre directia respectiva.

Eu m-am gandit a rezolva algoritmul in urmatorul fel: sunt intr-un labirint cu ochii inchisi, iar singura modalitate de a iesi este de a tine mana pe peretele drept. Ultima parte a codului face referire strict la aceasta treaba: in momentul in care peretele drept este liber, schimb directia in jos samd. Algoritmul se termina in momentul in care atinge o margine.