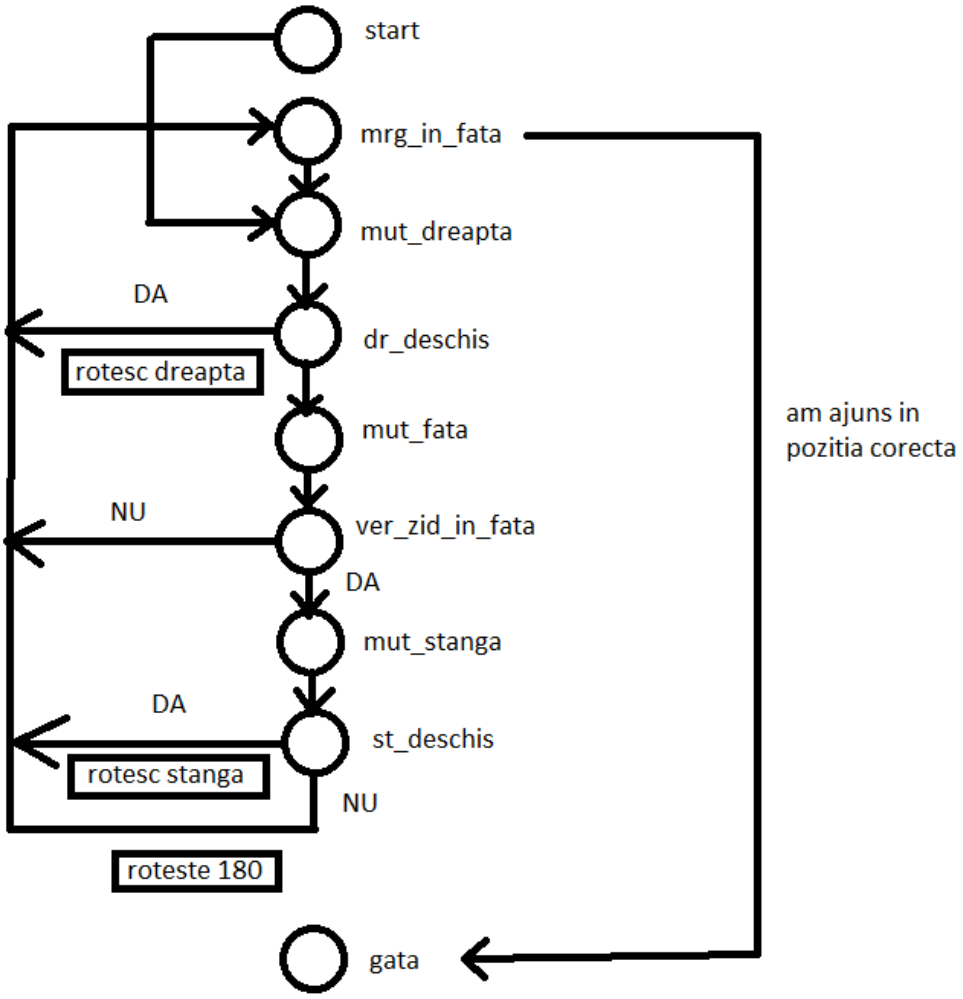
In cadrul proiectului meu am ales un automat destul de simplu. Automatul pentru algoritmul Wall Follower Right facut de mine este urmatorul.



Tin sa mentionez ca variabila mea directie functioneaza in felul urmator. Ea poate lua valorile 0 (jos), 1 (dreapta), 2 (sus), 3 (stanga). Aceasta variabila directie este luata ca si cand ma uit la matrice de sus. Valorile vor ramane asa mereu, iar directie va fi schimbat in functie de orientarea mea. Un exemplu de cum functioneaza variabila directie ar fi: daca eu am directia la 0 si vreau sa o schimb la dreapta atunci voi schimba variabila la 1, dar daca am varibila directie egala cu 1 si vreau sa ma mut iar la dreapta o voi schimba la 2.

Ideea de baza este urmatoarea: fiind Wall Follower Right voi verifica mereu ce am in dreapta, daca este liber imi voi schimba directia spre dreapta si ma voi muta, daca nu voi verifica in continuare ce am in fata. Daca am liber in fata ma voi misca si daca nu, atunci voi verifica ce am in stanga mea. La fel ca la celelalte stari, daca

nu am nimic, atunci ma voi roti la stanga si ma voi muta, dar daca am zid, atunci ma voi intoarce la 180 grade (deoarece directia a ramas aceeasi de la inceputul verificarii si cum am zid si in dreapta si in fata si in stanga atunci singurul lucru pe care il pot face este sa ma intorc in spate).

Pasii implementati sunt urmatorii:

- Incep cu o stare de ***start*** (pozitia initiala) in care initializez row si col cu ce am eu initial, initializez variabila directie cu 0 (jos) iar next_state va fi mut_dreapta
- Starea ***mut_dreapta*** este o stare in care doar voi face operatii in pe variabilele col sau row, astfel incat sa ma mute la dreapta. Operatiile sunt facute relativ la orientarea mea Am un case pentru fiecare directie (0, 1, 2, 3). Spre exemplu daca variabila directie este 0 atunci voi face *col + 1*, daca directia mea este 1 atunci voi face *row – 1* etc. Acelasi lucru este valabil pentru toate starile mele, voi avea un case in functie de directie, iar operatiile se vor face in functie de directia curenta. Am aceasta stare de mutare a col sau a row pentru ca am nevoie in starea urmatoare de aceasta variabila pentru a verifica daca am zid sau nu, asa ca la sfarsit dau maze_oe = 1. De asemenea salvez coorodnatele vechi in row_aux si col_aux.
- Starea ***dr_deschis*** este o stare in care verific daca am zid in pozitia in care am fost mutat in starea mut_dreapta. Daca este deschis atunci voi merge in starea mrg_in_fata, dar nu inainte sa resetez coordonatele mele la pozitia in care ma aflam cu ajutorul auxiliarelor. Daca nu este deschis atunci voi merge in mut_fata.
- Starea ***mrg_in_fata*** este o stare care ma deplaseaza in directia in care sunt. Daca sunt in directia 0 imi face row + 1, daca sunt in 1 atunci face col + 1 etc. De asemean imi pune maze_we pe 1 pentru a scrie in loc de 0 2 si sa stiu ca am trecut pe acolo. De asemenea aici verific daca am ajuns la finalul labirintului prin verificarea randului sau a coloanei (daca sunt 0 sau 63), daca am ajuns, voi trece in starea gata.
- Starea ***gata*** ii atribuie lui done valoarea 1 semn ca am terminat.
- Celelalte stari functioneaza in exact acelasi mod, conform automatului de mai sus. Starile cu mut, doar imi vor modifica pozitia pentru a ma putea folosi de variabila maze_in si sa verific daca am zid sau nu (1 sau nu), uar starile ver_zid_in_fata sau st_deschis fac verificarea efectiva si in functie de rezultat ma voi roti (ca mai apoi sa ma mut cu ajutorul starii mrg_in_fata) sau ma voi duce in starea urmatoare (ca in schema de mai sus).