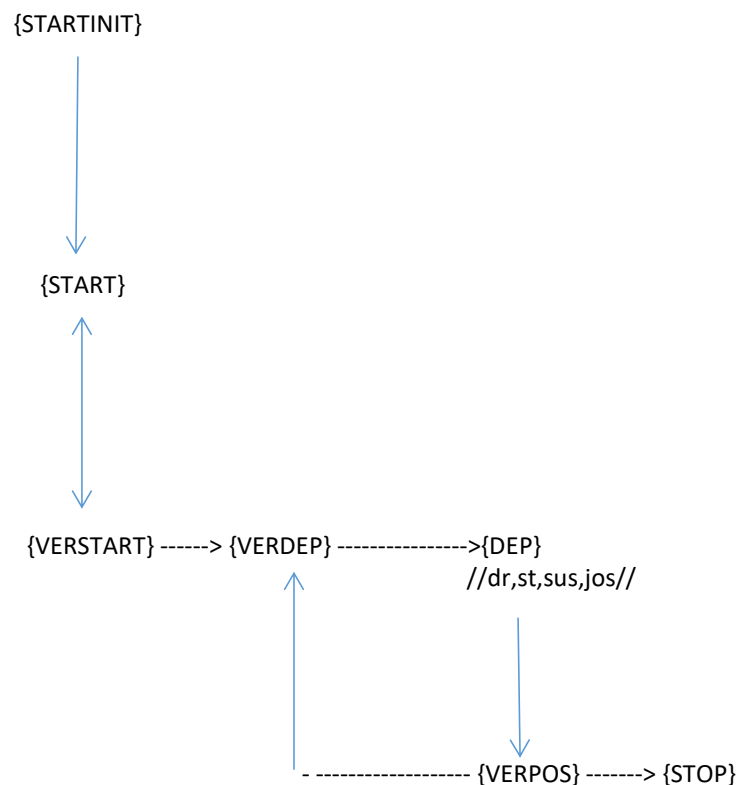


Pentru implementare, am realizat urmatorul automat de stare:



Explicarea starilor si a algoritmului:

STARTINIT: -> se preiau coordonatele initiale si se atribuie coordonatelor de deplasare, dar si copiilor acestora;

-> activez maze\_we pentru a pune 2 in pozitia initiala;

-> setez deplasarea initiala cu dreapta(0).

START si VERSTART:

-> le folosesc pentru a gasi plecarea din start;

-> se vor apela una pe cealalta pana cand o vor gasi;

VERDEP,STARILE DE DEPLASARE(depDr, depSt, depJos, depSus) si VERPOZ:

-> ele reprezinta starile principale de deplasare prin labirint;

-> in starea VERDEP voi verifica, in functie de directia de deplasare curenta, ce am in dreapta mea(dreapta difera si ea in functie de cazul de deplasare) pentru a vedea daca am zid sau spatiu: daca am zid voi pastra directia de deplasare, iar daca am spatiu voi schimba aceasta directie in urmatoarea stare.

-> in STARILE DE DEPLASARE, in functie de rezultatul dat de verdep, voi sti daca pastrez starea de deplasare sau o schimb

(in functie de ce am gasit in dreapta mea). In aceste stari se va realiza si deplasarea propriu zisa;

-> verpoz imi indica ce valoarea am in starea mea curenta, astfel voi citi din maze\_ in valoarea intoarsa de STARILE DE DEPLASARE.Daca voi gasi 0, voi verifica daca ma aflu pe marginea labirintului.Daca DA,marchez cu 2 pozitia, trec in STOP si

fac DONE = 1.Daca voi gasi un 1, ma voi reintoarce in starea de unde am venit si voi schimba cazul de deplasare printr-o rotire de 180 de grade(cazurile de deplasare sunt complementare: daca eu ma duceam spre dreapta si am gasit zid in fata,voi trece la cazul de stanga, intrucat spre stanga voi verifica daca am zid sus(la dreapta verificam jos).Astfel, in functie de ce gasesc voi sti care este noua mea directie de deplasare in verdep. Explicatia este similara si pentru cazul sus-jos.

-> verpoz o voi folosi si la marcarea cu 2 a drumului meu.