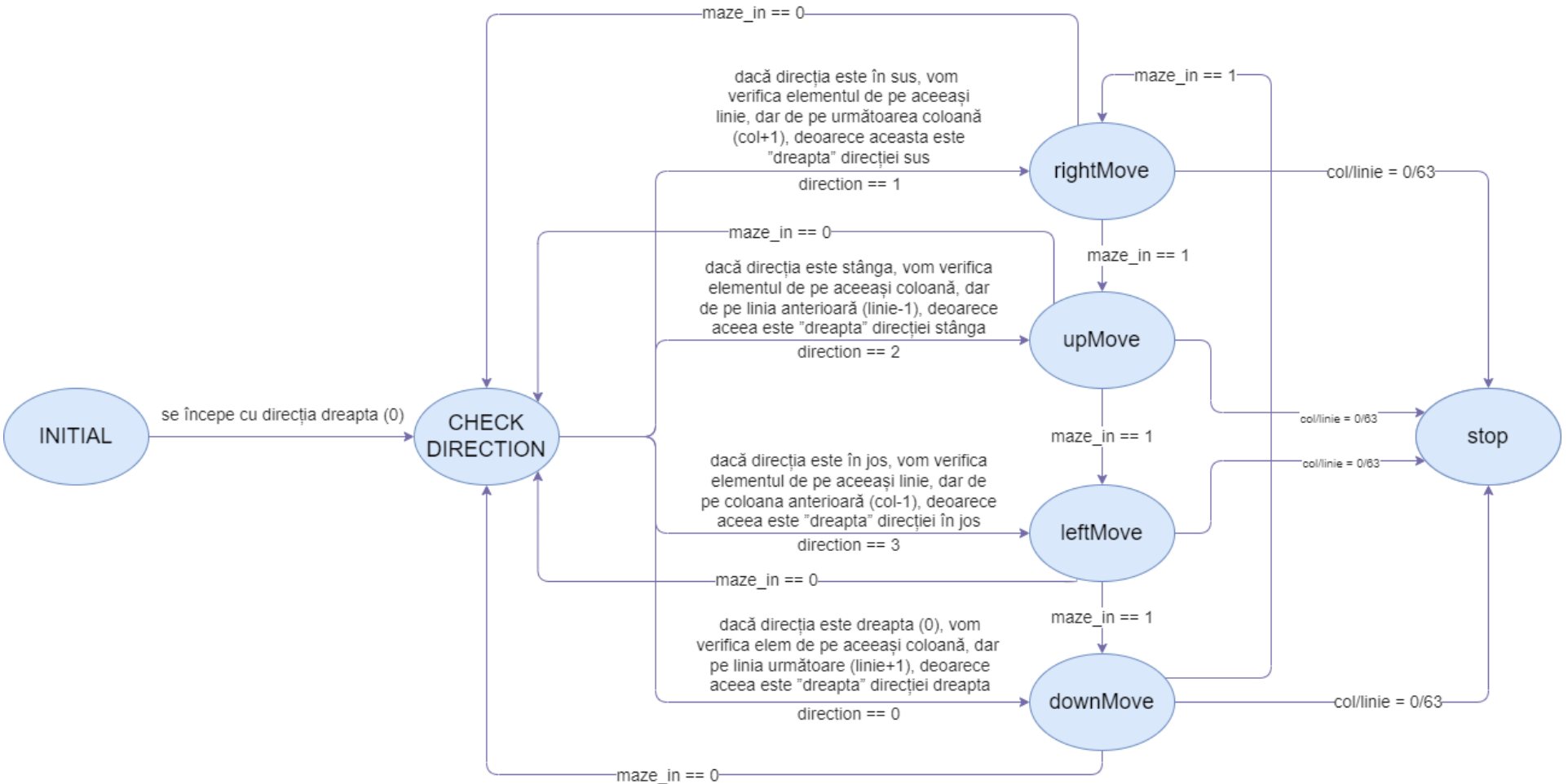
Sima Maria-Cristina
Grupa 331AB

TEMA 2 – README



Întâi am realizat automatul (cel din figura de mai sus), cu următoarele stări: initial, checkDirection, rightMove, upMove, leftMove, downMove și stop. Variabila ”direction” definește direcțiile în care poate fi parcurs labirintul, asemănător punctelor cardinale (0 – dreapta/est, 1 – sus/nord, 2 – stânga/vest, 3 – jos/sud), iar ordinea lor este legată de ordinea de parcurgere a verificărilor (întâi mereu verificăm la dreapta în raport cu direcția în care ne ”aflăm”, apoi continuăm cu rotirea spre dreapta).

DESCRIEREA STĂRILOR

1. Initial

În starea inițială, am declarat row și col cu starting_row și starting_col, pentru a începe rezolvarea labirintului de la punctul de start. Am adăugat și două variabile auxiliare, current_row și current_col, care ajută la mutarea în labirint pe anumite poziții, pentru a putea verifica în următoarele stări. De asemenea, am dat valoarea 0 variabilei direction și, astfel, verificarea se începe din direcția dreapta.

2. checkDirection

În această stare, am dat valori lui row și col în funcție de direcția în care dorim să verificăm. De exemplu, dacă direcția este în sus (numărul 1 din cod/nord), vom verifica în dreapta direcției în sus, adică pe aceeași linie, dar pe următoarea coloana (col = current_col + 1). Deoarece dorim să ne ”deplasăm” spre dreapta, next_state va lua valoarea stării rightMove, pentru a fi realizată verificarea pentru acea coordonată. Similar se realizează verificarea și pentru celelalte direcții. Maze_we va fi inactiv, pentru că nu dorim să scriem ”2” în niciun loc, iar maze_oe este activ, deoarece dorim să returnăm pe maze_in informația de la coordonatele [row, col].

3. rightMove, upMove, leftMove, downMove

Am realizat aceste stări pentru verificarea în direcția corespunzătoare a punctului în care ne situăm. De exemplu, dacă direcția este stânga (numărul 2 din cod/vest), vom dori să verificăm ”dreapta” acestei direcții, adică în sus. Așadar, din checkDirection, next_state va lua valoarea stării upMove. În upMove, verificăm întâi dacă maze_in este egal cu 0 și, în caz afirmativ, înseamnă că am găsit cale liberă și, deci, ne vom putea muta în acele coordonate. Direcția se va schimba (deoarece, practic, ne vom ”uita” în sus/la nord, deci direction va lua valoarea 1), maze_we se va activa, pentru că dorim să scriem 2 în locul în care am fost, și vom reveni înapoi la starea de checkDirection, intrând in if-ul pentru direction == 1 din checkDirection (deoarece s-a schimbat direcția). De asemenea, în cazul în care maze_in este 0 și se poate realiza mutarea pe poziția respectivă, trebuie să fie făcută verificarea dacă s-a ieșit din labirint sau nu. Așadar, am introdus aici o instrucțiune de tip if, care verifică index-ul lui col și lui row și, dacă s-a ieșit din matrice, next_state va deveni stop.

În cazul în care maze_in este 1, înseamnă că am întâlnit un perete, și nu un spațiu gol pe care ne-am putea deplasa. Astfel, vom continua verificările, vom schimba valorile lui row și lui col și vom activa maze_oe, deoarece dorim în continuare să citim datele de la pozițiile următoare. Aceste stări se află într-un ”ciclu”, până se găsește poziția pe care ne vom putea deplasa/ prima poziție care nu are perete.

Similar se procedează și pentru rightMove, leftMove și downMove.

4. Stop

Această stare indică faptul că s-a terminat parcurgerea labirintului și, în acest sens, semnalul done se va activa.