**Algoritmul Minimax în jocul X-0**

Pentru a aplica algoritmul minimax în jocurile cu doi jucători, vom presupune că X este un jucător de maximizare și O este un jucător de minimizare. Jucătorul care maximizează va încerca să își maximizeze scorul sau cu alte cuvinte va alege mișcarea cu cea mai mare valoare. Jucătorul care minimizează va încerca să minimizeze valoarea jucătorului care maximizează, alegând astfel mișcarea cu valoarea minimă.

Pentru a calcula valorile menționate mai sus, trebuie să decidem asupra unor ipoteze. Aceste valori le numim valori euristice.

În tic-tac-toe avem 3 posibilități:

Starea tabloului este o remiză: vom da acestei tablouri o valoare de 0;

X câștigă într-un stat de consiliu: îi vom oferi acestui consiliu o valoare de 100;

O câștigă într-un stat de consiliu: vom acorda acestei consilii o valoare de -100;

**Exemplificare algoritm**

Pentru a ilustra mai mult algoritmul minimax, să aruncăm o privire vizuală. În diagrama de mai jos, luați în considerare o situație în care este rândul lui X având în vedere starea actuală. X are trei mișcări posibile:

Nivelul 1: X are trei mișcări posibile și încearcă să găsească nodul maxim.

Nivelul 2: Prima mișcare duce la câștig direct pentru X, astfel se acordă 100 de puncte.

Nivelul 2: a doua și a treia mișcare vor duce la alte două mișcări posibile acolo unde este rândul lui O.

Nivelul 3: O încearcă să minimizeze scorul, astfel încât alege nodurile cu valoarea minimă.

Nivelul 3: Prima mișcare pentru O va duce la o victorie și a doua la o remiză, astfel presupunem că O va alege prima mișcare și nodul părinte va avea o valoare de -100. La fel și pentru a treia și a patra mișcare.

Înapoi la nivelul 1, X trebuie să aleagă acum între 100, -100 și 0. Deoarece X este maximizatorul, va alege cu siguranță 100, ceea ce va duce la o victorie.

După cum puteți observa, propagăm recursiv arborele de posibilități calculând scorul pentru fiecare stare terminală și apoi revenind pentru a decide ce mișcare vom lua.

**NEXT SLIDE**

Acum imaginați-vă o situație în care X poate câștiga în două moduri posibile, dar într-un mod este nevoie de mai multe mișcări decât celălalt. Dacă urmărim implementarea noastră actuală, ambele mișcări vor întoarce un scor de 100. Apoi vom alege aleatoriu mișcarea; cu toate acestea, ar fi mai bine dacă alegem direct cea mai scurtă cale de a câștiga.

Pentru a aborda acest lucru, vom scădea adâncimea sau nivelul curent din scorul tabloului în cazul în care jucătorul este maxim sau adăugăm adâncimea scorului în cazul în care este un jucător minim. În acest fel, pentru un jucător la maxim, calea mai scurtă va avea un scor mai mare, deoarece o adâncime mai mică este scăzută din acesta și invers pentru un jucător la minim.

Acest mod va ajuta, de asemenea, să pierdeți într-un mod mai lung dacă există o modalitate mai scurtă de a pierde. Iată un exemplu vizual după adăugarea adâncimii:

În cazul de mai sus, X va alege în mod evident 99 peste 97, deoarece este o modalitate mai ușoară de a câștiga.