

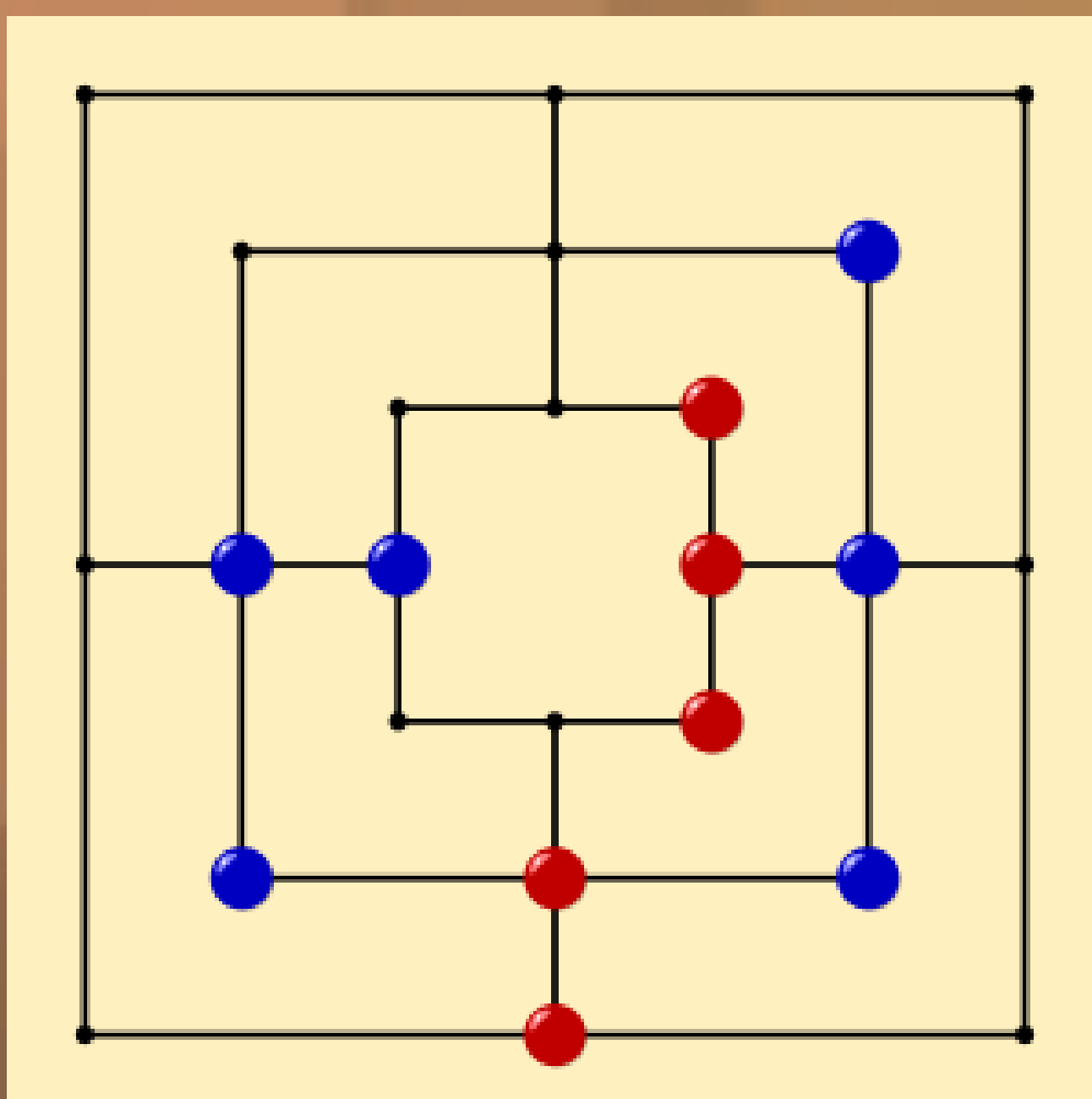
MICA

Jovo Šunjka SW17/2015, Jovana Jovanović SW26/2015
Fakultet tehničkih nauka, Novi Sad

Uvod

U ovom radu ćemo pisati o enginu za igru Mica, koji će podržavati igru dva igrača jedan protiv drugog, a igrači mogu da budu čovek, Minimax algoritam sa Alpha-Beta pruningom i Reinforcement learning.

Performanse će biti merene samo u slučaju kada igraju algoritmi. U sekciji rezultati su objavljeni dobijeni rezultati.



Igra Mica

Mica je igra koju igraju dva igrača, jedan protiv drugog. Svaki od igrača na početku igre ima po 9 figura na tabli. Tabla se sastoji od tri kruga sa po 8 tačaka. Cilj igre je da igrač spoji 3 figure u nizu, odnosno micu. Kada igrač spoji micu, onda mu je dozvoljeno da protivniku "pojede" jednu figuru. Pobjednik je igrač koji protivnika svede na 2 figure ili ga dovede u situaciju da ne može da pomeri nijednu figuru.

Igra se sastoji od 3 faze.

Faza 1 [FAZA POSTAVLJANJA]: Protivnici naizmenično postavljaju figure na slobodne tačke.

Faza 2 [FAZA POMERANJA]: Igrači naizmenično pomeraju figure po tabli

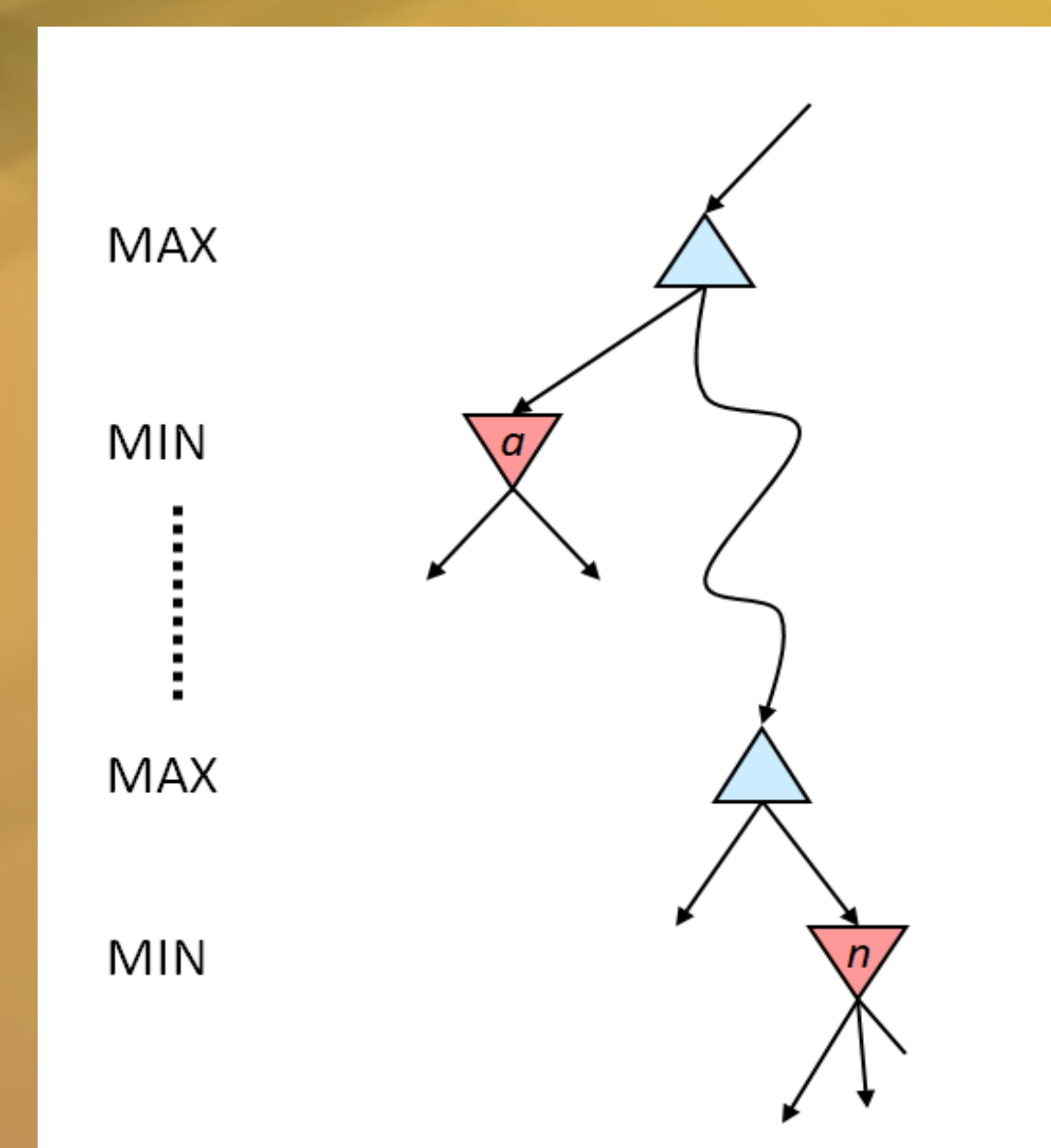
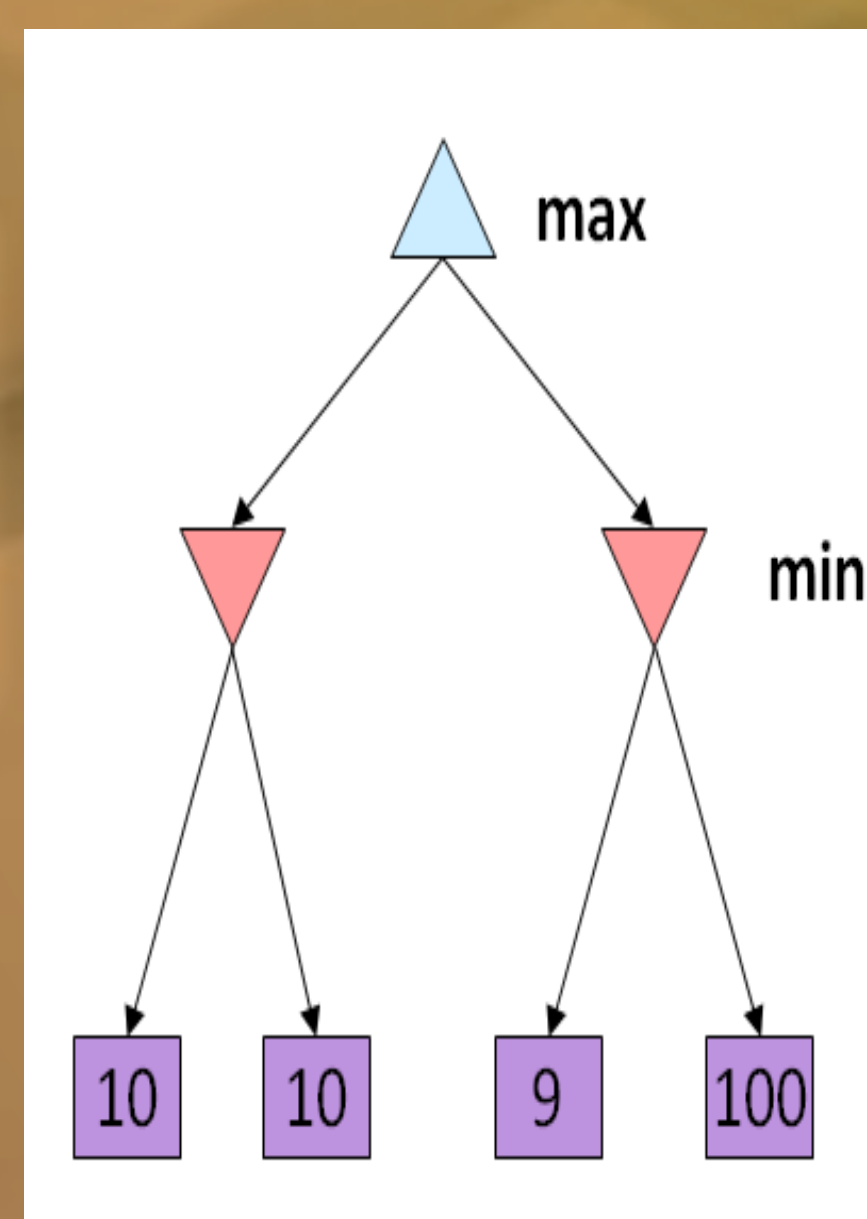
Faza 3 [FAZA SKOKA]: Počinje kada jedan od igrača spadne na 3 figure. U ovoj fazi igrač može da skače na slobodne tačke.

Minimax algoritam

Spada u grupu algoritama koji se koriste za pretragu sa protivnicima. Koristi stablo pretraživanja sa stanjima sveta. Za svaki čvor u stablu računamo njegovu minimax vrednost, odnosno najbolju moguću korisnost koju možemo dobiti iz tog čvora uz pretpostavku da imamo protivnika koji igra optimalano. Za stanja koja su pod kontrolom protivnika uzimamo minimalnu vrednost, a za stanja koja su pod kontrolom igrača maksimalnu vrednost.

Karakteristike:

- Vremenska složenost: $O(b^m)$
- Prostorna složenost: $O(bm)$
- Optimalana – samo kada i protivnik igra optimalno.

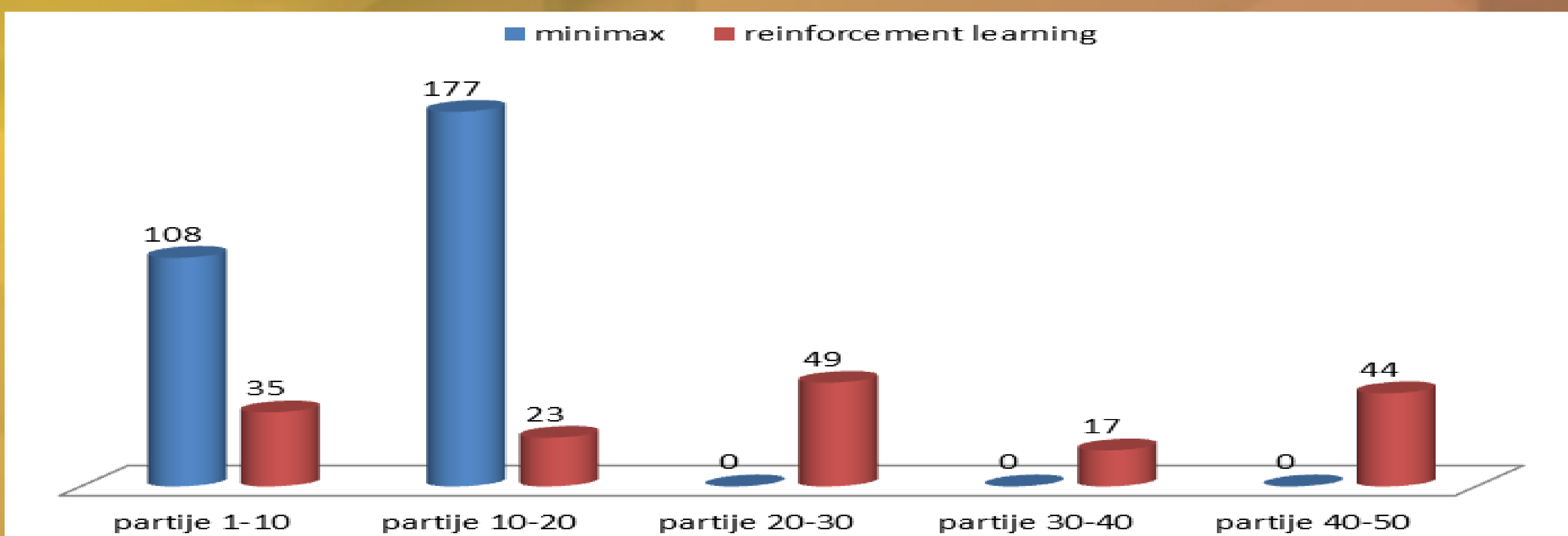
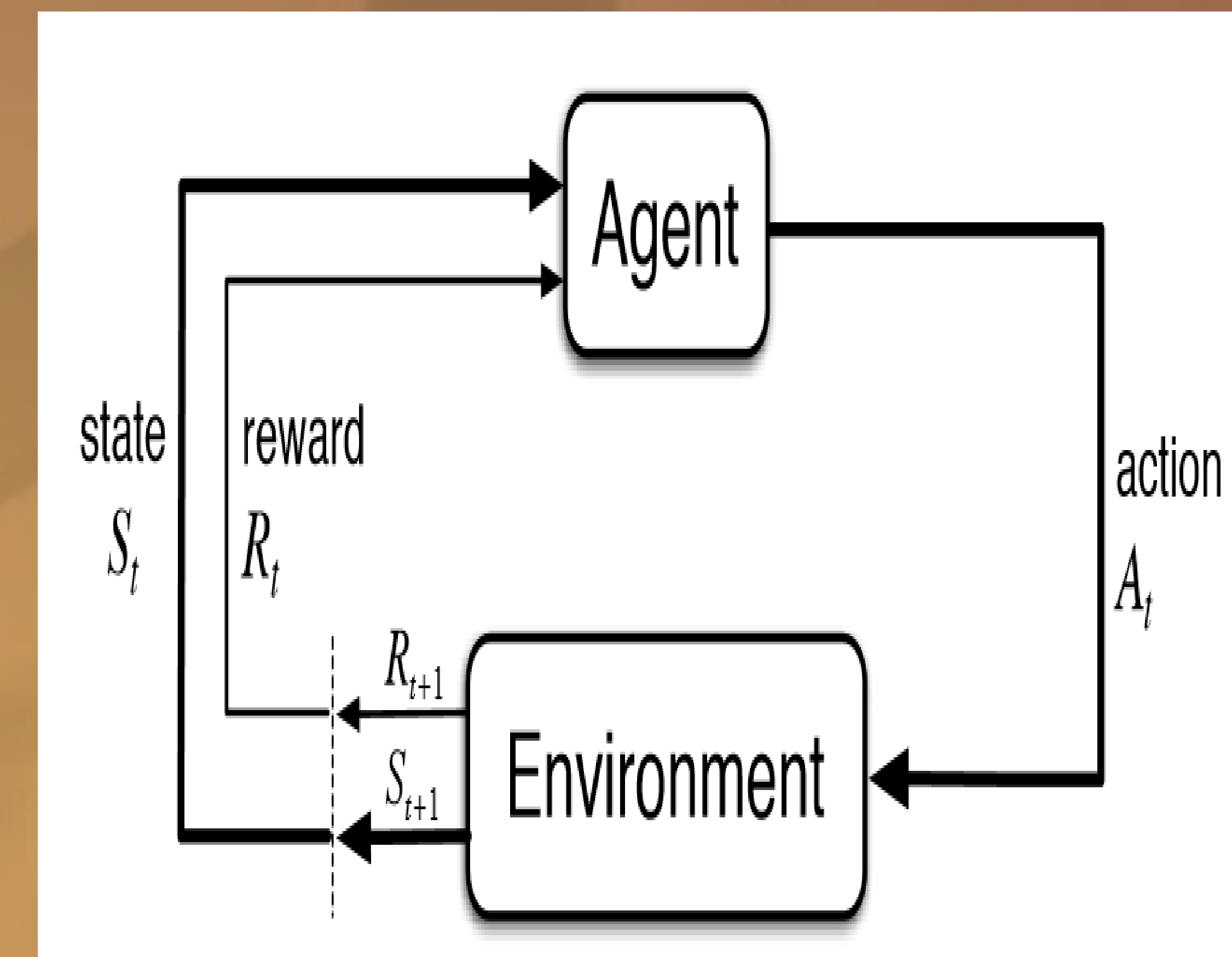


Alpha-Beta pruning

Koristimo ga da bi smo poboljšali performanse minimax algoritma. Prvo izračunamo min vrednosti za neku čvor n , a zatim prođemo kroz sve njegove protivnike. Min vrednost od čvora n nam je važna za MAX deo. Neka je a najbolja vrednost koju max može da dobije sa trenutne putanje. Ako vrednosti u n postanje lošija od a , MAX nikad neće birati n , pa možemo da izbacimo čvor n i njegove potomke.

Reinforcement learning

Reinforcement learning (RL) je oblast mašinskog učenja fokusirana na obučavanje agenata, koji kroz interakciju sa svojim okruženjem, treba da postignu odgovarajući cilj. Agent bira i izvršava akcije koje utiču na njegovo okruženje, od kog kao povratnu informaciju dobija podatke o novom stanju i nagradi. Nagrada agentu govori koliko je bila pogodna izabrana akcija u datom stanju okruženja. Cilj agenta je da bira akcije na takav način da se maksimizuje kumulativna nagrada koju dobija u procesu odlučivanja.



Rezultati

Posle 50 odigranih partija između minimax algoritma i reinforcement learning-a dobili smo sledeće rezultate. Radi lakšeg pregleda uzeli smo prosek odigranih poteza do kraja igre posle svakih 10 igara.

Minimax algoritam pobeđuje samo 3 puta i to 2 u prvih 10 igara i 1 u sledećih 10 igara,

Zaključak

Na osnovu dobijenih rezultata možemo zaključiti da minimax algoritam nije najbolje implementiran i da kako se povećava broj odigranih partija igrač koji je obučavan sa reinforcement learningom igra sve bolje i bolje. Glavni problem kod reinforcement learning-a je ogroman broj novih stanja s kojima se upoznaje u Mici. Na tabli se nalaze 24 polja, a svake polje može zauzeti plavi ili crveni igrač, ili može ostati prazno polje. To znači da postoje $3^{24} \approx 282$ milijarde kombinacija. Ralph Gasser je pokazao da postoji veliki broj stanja koja su simetrična, tako da je maksimalan broj jedinstvenih stanja 10^{10} , što je i dalje velik broj.

Usled nedevojne obučenosti, algoritam još nije spreman da pobeđi eksperte, ali biće. USKORO!