

SBNZ - Predlog projekta: Poker Player Expert System

Članovi tima: Luka Petković SV 16/2021, Uroš Muškinja SV 77/2021

Opis problema koji se rešava

1. Motivacija

Poker je kompleksna kartaška igra koja zahteva računanje verovatnoća, psihološku analizu protivnika i strateško razmišljanje. Iako su pravila igre relativno jednostavna, donošenje optimalnih odluka u realnom vremenu je težak izazov čak i za najiskusnije igrače. Upravo zato, postoji potreba za sistemom koji može da analizira trenutnu situaciju na stolu, prati ponašanje protivnika kroz vreme, i predloži najbolji potez u datom trenutku. Ovaj sistem bi bio od velikog značaja kako za novije igrače, tako i za one koji su već savladali osnovne strategije igre.

2. Pregled problema

Konkretan problem koji se rešava je automatska analiza poker situacije i preporučivanje optimalnog poteza na osnovu trenutnog stanja igre, istorije poteza i karakteristika protivnika. Nijedan sistem za preporuku ne može garantovati pobedu u svakoj partiji pokera (jer je sreća faktor u ovoj igri), ali može znatno povećati šanse za to.

Trenutno postoje različiti pristupi rešavanja ovog problema:

- Poker kalkulatori: efikasni, ali se fokusiraju samo na matematičke verovatnoće na osnovu karata (ne uzimaju u obzir prethodnu istoriju poteza protivnika)
- GTO (Game Theory Optimal) solveri: računaju teorijski optimalne strategije ali takođe ne uzimaju u obzir specifično ponašanje protivnika
- AI sistemi poput Libratus/Pluribus - napredni ali previše kompleksni, teški za razumevanje i primenu

Prednosti našeg rešenja:

- Kombiniuje matematičke proračune sa analizom ponašanja: ne oslanja se samo na verovatnoću
- Transparentno rezonovanje: korisnik može da vidi zašto je određena odluka preporučena
- Adaptivno učenje: sistem uči obrasce ponašanja protivnika kroz CEP

3. Metodologija rada

3.1. Očekivani ulazi u sistem (input)

- Trenutno stanje igre: karte igrača, karte na stolu, trenutni pot, pozicija igrača na stolu, broj aktivnih protivnika
- Informacije o protivnicima: stack size, pozicija i potezi svakog protivnika
- Istorijske informacije: prethodne akcije protivnika u ranijim rundama, njihovo ponašanje (npr. koliko često "blefiraju")
- Kontekstne informacije: faza igre (pre-flop, flop, turn, river), pozicije SB (small blind) i BB (big blind)

3.2 Očekivani izlazi iz sistema (output)

- Preporučeni potez: Fold, Call, Raise (sa preporučenom sumom), All-in
- Obrazloženje odluke: matematičko opravdanje, strategijski razlozi
- Dodatne informacije: verovatnoća pobeđe trenutne ruke, preporučena buduća strategija itd.

3.3 Baza znanja projekta

- Matematičke tabele: hand ranking, equity tabele, pot odds kalkulacije
- Strateška pravila: pozicijska strategija (ranija/srednja/kasnija pozicija), pravila za različite faze igre
- Profili protivnika: kategorija protivnika u zavisnosti koliko voli da rizikuje (Tight, Loose, Aggresive, Passive), ponašanje za svaku kategoriju

Popunjavanje baze znanja:

- Inicijalno: unesena teorija i pravila pokera, jačine ruku, kao i matematičke formule
- Dinamički: CEP komponenta će kroz vreme učiti obrasce ponašanja protivnika

Interakcije na osnovu znanja:

1. Forward chaining (od podataka ka zaključku) - koristi se za određivanje osnovnih strategija na osnovu dostupnih podataka.

Primer:

Ulaz: ruka igrača = A♠K♠, pozicija = "kasna", broj aktivnih igrača = 2.

Pravila: 1) ako igrač ima *premium* ruku (AA, KK, QQ, AK), onda se klasifikuje kao jaka startna ruka.

2) ako je ruka jaka i igrač je u kasnoj poziciji, preporučena strategija je *raise*.

3) ako je preporučena strategija *raise* i broj aktivnih igrača ≤ 3 , sistem predlaže *all-in*

Zaključak: sistem predlaže *all-in* jer su uslovi zadovoljeni

2. Backward chaining (od cilja ka uslovima)

Primer: Da li bi igrač trebalo da blefira?

Cilj: "Da li treba blefirati?" - da ako važe podcilj 1 i podcilj 2

Podcilj 1: da li protivnik verovatno folduje? (OpponentLikelyToFold)

- da ako važe 1.1 (protivnik je tight) i 1.2 (Board looks scary)

Podcilj 2: da li igrač ima slabu ruku (MyHandIsWeak)

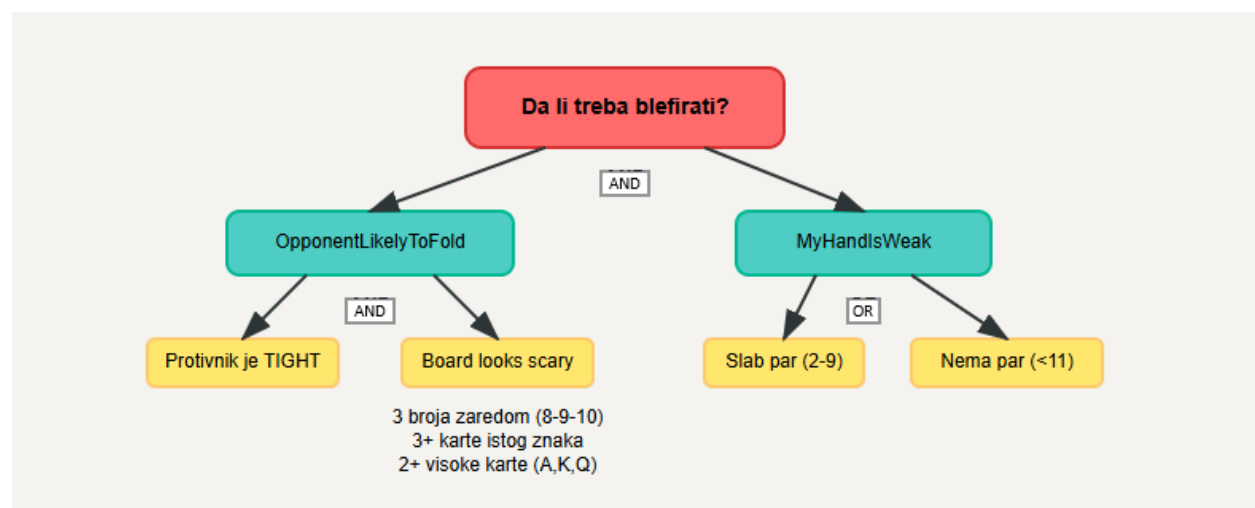
- da ako važi 2.1 (slab par) ILI 2.2 (nema ni par)

Pod-podcilj 1.1: profil protivnika (kategorija) je "TIGHT"

Pod-podcilj 1.2: Board looks scary – na tabli su takve karte da je vrlo moguć *straight* (na stolu se nalaze 3 broja zaredom npr. 8-9-10), ili *flush* (3 ili više karti istog znaka), ili ako su na stolu bar 2 "visoke" karte (A,K,Q)

Pod-podcilj 2.1: igrač ima samo par, i to slab (između para dvojki i para devetki)

Pod-podcilj 2.2: igrač nema ni par, odnosno samo visoku kartu (High Card) manju od 11 (J)



3. CEP (analiza kroz vreme) - koristi se da sistem prepozna obrasce ponašanja protivnika

Primer 1: Bluff Detection

Ovaj pattern prati igrače koji često blefiraju u kratkom vremenskom periodu. Kada sistem detektuje da je igrač blefirao 3 ili više puta u poslednjih 5 minuta, klasifikuje ga kao “čestog blefera”. Ta informacija se kasnije koristi u odlučivanju poteza – protiv čestih blefera možemo češće da ispratimo njihove *bet*-ove jer verovatno nemaju jaku ruku.

Primer Drools koda:

```
rule "Detect Frequent Bluffer"
```

```
when
```

```
    $player: Player()
```

```
    Number(intValue >= 3) from accumulate(
```

```
        BluffEvent($p: player, this after[0s,5m] $start) from entry-point "player-actions"
        and $start: BluffEvent(player == $p),
```

```
        count($p)
```

```
    )
```

```
then
```

```
    insert(new PlayerProfile($player.getId(), "FREQUENT_BLUFFER", 0.7));
```

```
    System.out.println("Player " + $player.getId() + " classified as frequent bluffer");
```

```
end
```

Pravilo kako treba igrati protiv blefera:

```
rule "Call Against Frequent Bluffer"
```

```
when
```

```
    $situation: GameSituation(handStrength > 0.25) // relativno slaba ruka
```

```
    PlayerProfile(playerId == $situation.opponentId, profileType == "FREQUENT_BLUFFER")
```

```
    $potOdds: PotOdds(ratio < 0.35) // trebamo samo 35% win rate
```

```
then
```

```
    $situation.setRecommendedAction("CALL");
```

```
    $situation.setReason("Opponent is frequent bluffer - calling with lower threshold");
```

```
end
```

Primer 2: Tilt Detection Pattern

Tilt je psihološko stanje igrača kada postaje emocionalan nakon velikog gubitka i počinje da igra iracionalno. Ovaj pattern detektuje tilt tako što prati da li je igrač doživeo "bad beat" (veliki gubitak sa jakom rukom), a zatim u narednih 5 minuta ima više od 2 agresivna poteza i prosečan bet 2x veći od njegovog normalnog. Igrač na tiltu je laka meta jer donosi loše odluke.

Primer Drools koda:

```
rule "Detect Player on Tilt"
```

```
when
```

```
    $player: Player()
```

```
    BadBeatEvent(player == $player, $lossTime: timestamp) from entry-point "game-events"
```

```
    Number(intValue >= 3) from accumulate(
```

```
        AggressiveActionEvent(player == $player,
```

```
        this after[0s,5m] $lossTime) from entry-point "player-actions",
```

```
        count()
```

```
    )
```

```
    Number(doubleValue > $player.avgBetSize * 2) from accumulate(
```

```
        BetEvent(player == $player, $amount: amount,
```

```
        this after[0s,5m] $lossTime) from entry-point "player-actions",
        average($amount)
```

```
    )
```

```
then
```

```
    insert(new TiltIndicator($player.getId(), "HIGH_TILT", now()));
```

```
    System.out.println("Player " + $player.getId() + " is on TILT - exploit opportunity!"); end
```

Pravilo kako treba igrati protiv igrača na tiltu:

```
rule "Exploit Tilted Player"
```

```
when
```

```
    $situation: GameSituation(handStrength > 0.4) // srednja ruka
```

```

TiltIndicator(playerId == $situation.opponentId, level == "HIGH_TILT")

then

    $situation.setRecommendedAction("RAISE");

    $situation.setReason("Opponent is on tilt - exploit with wider range");

end

```

Napomena: ove Drools implementacije su demonstrativne i podložne promjenama ukoliko se ispostave kao neadekvatne. Oba patterna koriste temporal window od 5 minuta, povezuju više događaja, i koriste accumulate funkciju.

4. Template (rule based) - umesto da za svaku početnu ruku *hardcode*-ujemo preporučene akcije za svaku fazu igre, možemo koristiti sliku ispod, i napraviti template fajl (.drt ili .csv) koji bi učitali i koristili u pravilima. Na ovaj način, lako se dodaju nove kombinacije bez izmene koda, i takođe svaka izmena u template-u utiče na sva pravila.

Texas Hold'em Starting Hands

A-A	A-K	K-Q	Q-J	J-T	T-9	9-8	8-7	7-6	6-5	5-4	4-3	3-2
K-K	A-Q	K-J	Q-T	J-9	T-8	9-7	8-6	7-5	6-4	5-3	4-2	
Q-Q	A-J	K-T	Q-9	J-8	T-7	9-6	8-5	7-4	6-3	5-2		
J-J	A-T	K-9	Q-8	J-7	T-6	9-5	8-4	7-3	6-2			
T-T	A-9	K-8	Q-7	J-6	T-5	9-4	8-3	7-2				
9-9	A-8	K-7	Q-6	J-5	T-4	9-3	8-2					
8-8	A-7	K-6	Q-5	J-4	T-3	9-2						
7-7	A-6	K-5	Q-4	J-3	T-2							
6-6	A-5	K-4	Q-3	J-2								
5-5	A-4	K-3	Q-2									
4-4	A-3	K-2										
3-3	A-2											
2-2												

Playable

Playable to an extent

Playable until the first raise

Unplayable (Check/Fold Only)

PAIRS & SUITED CARDS

Primer template fajla (u csv formatu):

```

handType,card1,card2,suited,action,position
premium,A,A,false,RAISE,"EARLY MIDDLE LATE"
premium,K,K,false,RAISE,"EARLY MIDDLE LATE"
premium,A,K,true,RAISE,"EARLY MIDDLE LATE"
strong,A,K,false,RAISE,"MIDDLE LATE"
playable,A,Q,true,CALL,"LATE"
marginal,2,2,false,FOLD,"EARLY"

```

4. Konkretni primer rezonovanja

Scenario:

Igrač ima **K-pik Q-karo**. Flop: **Q-tref 7-herc 2-karo**. Protivnik betuje 500 u potu od 1000.

Koraci rezonovanja sistema:

1. Ulazni podaci

- Igračeve karte: K-pik Q-karo (top pair sa solidnim kickerom – srednje jaka ruka).
- Board: Q-tref 7-herc 2-karo.
- Pot: 1000 žetona.
- Bet protivnika: 500 žetona.

2. Jačina ruke

- KQ na flopu Q-tref 7-herc 2-karo - Top pair sa solidnim kickerom - klasifikuje se kao srednje jaka ruka.

3. Pozicija za stolom

- Igrač je u kasnoj poziciji → nastavljamo dalje (ne foldujemo odmah).

4. Profil protivnika

- Protivnik je **loose** → česta ulaganja i sa slabim rukama.

5. Win rate provera za loose protivnika

- Pot odds: treba platiti 500 da bi se osvojilo ukupno 2000 (25%).
- Procena šanse sa KQ protiv loose spektra $\approx 60\%$.
- $60\% > 40\%$ (prag za loose protivnike u dijagramu) → uslov zadovoljen.

6. Stack size provera

- Stack igrača je prosečan - nema dodatne prednosti za raise.

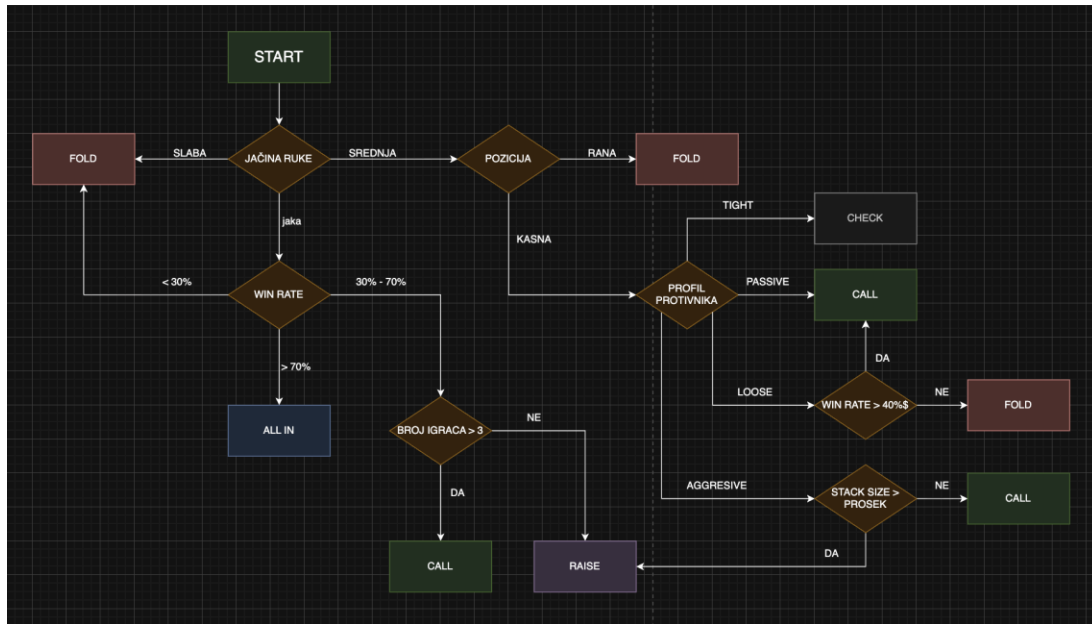
7. Broj igrača u ruci?

- Samo jedan protivnik - manji rizik, lakše donosimo odluku.

8. Zaključak (output)

- Preporučeni potez: **Call**.
- Obrazloženje: "Vaša ruka je top pair sa dobrim kickerom, protivnik verovatno ne igra samo jake ruke. Imate dovoljno dobru šansu da nastavite igru."

If/Else dijagram:



Decision table:

action	minEquity	maxEquity	position	stackRange
FOLD	0	30	any	any
CALL	31	50	early	any
RAISE	51	70	late	any
ALL-IN	71	100	any	small
RAISE	71	100	late	big

Napomena:

Ovo je samo demo primer jednog konkretnog poteza, sa samo dva igrača. U sklopu projekta će biti pokrivenne i situacije sa više poteza i situacije sa više igrača za stolom. Takođe, postoji mogućnost da ćemo uprostiti sistem ukoliko se ispostavi da je implementacija ovakvog sistema previše komplikovana.