Универзитет у Београду

Математички факултет

Филип Пешић

Коначни марковски процеси одлучивања с применом на игру Тексас холдем покер

Мастер рад

Београд, 2021.

Садржај

Садржај и циљ рада

У првом поглављу је приказана теорија која стоји иза метода учења условљавањем, почев од коначних марковских процеса одлучивања.

У другом поглављу су објашњена правила игре и наведени примери из рада програма.

Треће поглавље је посвећено стратегији успешног играња покера.

У четвртом поглављу се на основу стратегије формира модел компјутерског играча.

У петом поглављу је укратко описан програм.

У шестом поглављу су приказани резултати рада програма. Циљ рада је да добијени компјутерски играч има квалитетне перформансе.

1. Учење условљавањем, односно поткрепљивањем (енгл. Reinforcement learning)

Да бисмо постали добар возач аутомобила, није довољно да положимо возачки испит, већ је потребно искуство у вожњи. Оно се стиче интеракцијом са окружењем: Свака наша акција има своје последице. Опажањем различитих последица можемо научити шта треба да радимо да бисмо постигли жељени резултат. Такав приступ се може пренети на програме и зове се учење условљавањем или поткрепљивањем. Зове се тако зато што је то заправо метода покушаја и погрешака: последице се вреднују бројевима и представљају награде.

Дефиниција: Агент је особа или ствар која учи и доноси одлуке.

1.1. Коначни марковски процеси одлучивања

Нека је $\{X_t : t \in \mathbb{N}_0\}$ случајан процес.

Дефиниција: Случајан процес $\{X_t : t \in \mathbb{N}_0\}$ задовољава марковско својство ако важи: $P\{X_{n+1} = x \mid X_n = x_n, \dots, X_1 = x_1\} = P\{X_{n+1} = x \mid X_n = x_n\}$. ([1], глава 3.)

Дефиниција: Скуп стања $\mathcal{S}$ је скуп свих могућих вредности процеса $\{X_t : t \in \mathbb{N}_0\}$.

Дефиниција: Случајан процес $\{X_t : t \in \mathbb{N}_0\}$ се у тренутку t налази у стању $s \in \mathcal{S}$ ако је $X_t = s$.

Дефиниција: Акција $a \in \mathcal{A}(s)$ је нека радња услед које процес прелази у ново стање. $\mathcal{A}(s)$ је скуп свих могућих акција које је могуће изабрати док је процес у стању s.

Дефиниција: Награда $r \in \mathcal{R} \subset \mathbb{R}$ је вредност која се додељује процесу приликом избора акције $a \in \mathcal{A}(s)$ у стању $s \in \mathcal{S}$.

Проблем учења условљавањем се може приказати коначним марковским процесима одлучивања. Почнимо од једноставнијег, дискретног случаја.

Окружење се мења у временским тренуцима $t \in \mathbb{N}_0$. У сваком тренутку агент опажа окружење у стању $S_t \in \mathcal{S}$ и на основу тога бира акцију $A_t \in \mathcal{A}(s)$. Као последицу акције, агент добија награду $R_{t+1} \in \mathcal{R}$ и налази се у стању S_{t+1}. Процес и агент дају низ $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \dots$

У коначном марковском процесу одлучивања, случајне променљиве R_t и S_t имају дискретне расподеле које зависе само од претходног стања и акције (због марковског својства). Њихова расподела преласка је дефинисана вероватноћама:

$$p(s', r \mid s, a) = P\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$

за све $s', s \in \mathcal{S}$, $r \in \mathcal{R}$ и $a \in \mathcal{A}(s)$. Расподела потпуно описује процес. Одатле се може израчунати све остало, на пример:

расподела преласка:

$$p(s'|s, a) = \sum_{r \in \mathcal{R}} p(s', r \mid s, a)$$

очекиване награде за пар (стање, акција):

$$r(s,a) = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a)$$

1.1.1. Награда

Награда је начин да одредимо шта желимо да агент научи. На пример, могуће награде у покеру су колико новца је освојено/изгубљено, +1 за победу на турниру или +1 за сваку добијену руку. Јасно је да је једино „колико новца је освојено“ исправна награда, јер је циљ игре освојити што више новца. Награђивање других ствари доводи до скретања агента са правог пута и освајања мање новца него што је могуће. ([1], поглавља 3.2. и 3.3.)

Циљ агента је да максимизује укупну награду, тј. на дуже време. После тренутка t агент ће добити награде $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$ Желимо да њихова сума $G_t = r_{t+1} + r_{t+2} + r_{t+3} \ldots + r_T$, где је T последњи временски треунутак, буде максимална.

Процеси се деле на епизодичне и непрекидне, у зависности од тога да ли постоји последњи тренутак процеса. Епизодичан процес је подељен на независне подскупове – епизоде. Епизода је један цилкус рада процеса. После њеног завршетка, процес се „ресетује“ и започиње нови циклус, тј. нову епизоду.

Покер је епизодичан процес, а епизода је рука. Агент може да добије само једну награду и то приликом освајања новца.

1.1.2. Функција вредности и политика

Функција вредности стања говори колико је добро за агента да буде у одређеном стању, односно колико је добро да изабере акцију која води до тог стања, тј. колико је висока очекивана будућа награда. Наравно, награда зависи од будућих акција, па се функција вредности дефинише у односу на политику. ([1], поглавља 3.5. и 3.6.)

Дефиниција: Политика је пресликавање из скупова стања и акција у вероватноће избора свих могућих акција у сваком стању. Ако агент прати политику π у стању s, онда је $\pi(a|s) = P\{A = a \mid S = s\}$.

Различите методе учења дефинишу различите начине ажурирања политике као резултат искуства агента.

Дефиниција: Вредност стања s при политици π, $v_\pi(s)$ је очекивана награда процеса почев од стања s:

$$v_\pi(s) = \mathbb{E}_\pi(G_t \mid S_t = s) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} R_{t+k+1} \,\middle|\, S_t = s\right)$$

Вредност акције a у стању s је

$$q_\pi(s,a) = \mathbb{E}_\pi(G_t \mid S_t = s, A_t = a) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right)$$

Фундаментално својство функције вредности стања је рекурзивна веза са функцијом вредности следећих стања:

$$v_\pi(s) = \mathbb{E}_\pi(G_t \mid S_t = s) = \mathbb{E}_\pi(R_{t+1} + G_{t+1} \mid S_t = s)$$
$$= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + \mathbb{E}_\pi(G_{t+1} \mid S_{t+1} = s')\big)$$

$$= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_\pi(s')\big)$$

за све $s \in \mathcal{S}$.

Ово својство се назива Белманова једначина за v_π.

Дефиниција: Политика π је боља или једнака политици π', у ознаци $\pi \geq \pi'$, ако и само ако је $v_\pi(s) \geq v_{\pi'}(s)$ за све $s \in \mathcal{S}$.

Увек постоји бар једна политика која је боља или једнака свим осталим политикама (у сваком стању се бира акција са највећом вредношћу). То је оптимална политика. Означава се са π_* Оптималне политике деле исте оптималне функције вредности стања и вредности акције у стању: $v_*(s) = \max_\pi v_\pi(s)$ и $q_*(s,a) = \max_\pi q_\pi(s,a)$.

Како оптимална политика сугерише избор акције са највећом вредношћу, следи да је:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) = \max_{a \in \mathcal{A}(s)} \mathbb{E}_{\pi_*}(G_t \mid S_t = s, A_t = a) = \max_{a \in \mathcal{A}(s)} \mathbb{E}_{\pi_*}(R_{t+1} + G_{t+1} \mid S_t = s, A_t = a)$$

$$= \max_{a \in \mathcal{A}(s)} \mathbb{E}(R_{t+1} + v_*(S_{t+1}) \mid S_t = s, A_t = a) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_\pi(s')\big)$$

То је Белманова једначина оптималности за v_* (у даљем тексту Белманова једначина). Белманова једначина оптималности за q_* је:

$$q_*(s,a) = \max_{a \in \mathcal{A}(s)} \mathbb{E}(R_{t+1} + G_{t+1} \mid S_t = s, A_t = a)$$

$$= \max_{a \in \mathcal{A}(s)} \mathbb{E}(R_{t+1} + v_*(S_{t+1}) \mid S_t = s, A_t = a) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_\pi(s')\big)$$

За коначне марковске процесе одлучивања, Белманова једначина има јединствено решење независно од политике. То је заправо систем од n једначина са n непознатих, где је n број стања. Кад се пронађе v_*, оптимална акција је она која води ка стању са највећом вредношћу. Оптимална политика је свака политика која додељује позитивне вероватноће само оптималним акцијама. Можемо рећи, свака политика која је похлепна у односу на оптималну функцију вредности је оптимална политика.

Дефиниција: Политика π је похлепна у односу на функцију вредности q_π ако се у сваком стању $s \in \mathcal{S}$ увек бира акција a за коју је вредност $q_\pi(s,a)$ највећа.

Међутим, ретко је могуће директно решити Белманову једначину оптималности. Морају да важе следећи услови:

- тачно знамо динамику окружења;
- имамо довољну компјутерску снагу;
- марковско својство.

Сва три услова ретко важе и то у најједноставнијим случајевима. Зато се мора прибећи итеративним методама и желимо да нађемо политику која је довољно близу оптималне.

1.2. Динамичко програмирање

Динамичко програмирање је скуп алгоритама којима може да се израчуна оптимална политика ако се окружење у потпуности може описати марковским процесом одлучивања. Претпоставимо да је окружење коначан процес, тј. скупови стања и акција су коначни. ([1], глава 4.)

Алгоритми се добијају трансформацијом Белманових једначина у правила ажурирања за побољшање апроксимација функција вредности.

1.2.1. Рачунање политике

Нека је π произвољна политика. Белманова једначина је: ([1], поглавље 4.1.)

$$v_\pi(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)\big(r + v_\pi(s')\big)$$

v_π постоји и јединствена је за сва стања ако је процес епизодичан. Нека је $v_k, k \in \mathbb{N}_0$ низ функција вредности такав да је v_0 произвољна, а свака следећа се добија коришћењем Белманове једначине за v_π за претходну:

$$v_{k+1}(s) = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)\big(r + v_k(s')\big)$$

Под већ помеутим условом низ v_k конвергира ка v_π. Алгоритам се зове итеративна евалуација политике. Приликом сваке итерације ажурира се вредност сваког стања.

1.2.2. Побољшавање политике

Нека је π детерминистичка политика, v_π функција вредности стања и s стање. Желимо да знамо да ли је боље да у стању s увек бирамо акцију $a \neq \pi(s)$. Вредност праћења политике π је $v_\pi(s)$. Вредност бирања акције a је: ([1], поглавље 4.2.)

$$q_\pi(s,a) = \mathbb{E}_\pi(R_{t+1} + v_\pi(S_{t+1}) \mid S_t = s, A_t = a) = \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)\big(r + v_\pi(s')\big)$$

Ако је $q_\pi(s,a) > v_\pi(s)$, боље је у стању s бирати акцију a, а затим пратити политику π. Ово је специјални случај Теореме побољшавања политике.

Теорема: Нека су π и π' детерминистичке политике тако да је $q_\pi\big(s,\pi'(s)\big) \geq v_\pi(s)$ за свако $s \in \mathcal{S}$. Тада је $v_{\pi'}(s) \geq v_\pi(s)$ за свако $s \in \mathcal{S}$. Такође, за стања за која важи строга неједнакост у првом случају, важи и у другом.

Идеја доказа: ([2])

$v_\pi(s) \leq q_\pi\big(s,\pi'(s)\big)$

$= \mathbb{E}\big(R_{t+1} + v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(a)\big)$

$= \mathbb{E}_{\pi'}(R_{t+1} + v_\pi(S_{t+1}) \mid S_t = s)$

$\leq \mathbb{E}_{\pi'}\big(R_{t+1} + q_\pi\big(S_{t+1},\pi'(S_{t+1})\big) \mid S_t = s\big)$

$= \mathbb{E}_{\pi'}(R_{t+1} + \mathbb{E}_{\pi'}(R_{t+2} + v_\pi(S_{t+2})\ ) \mid S_t = s)$

$= \mathbb{E}_{\pi'}(R_{t+1} + R_{t+2} + \ v_\pi(S_{t+2}) \mid S_t = s)$

$\leq \mathbb{E}_{\pi'}(R_{t+1} + R_{t+2} + R_{t+3} + \ v_\pi(S_{t+3}) \mid S_t = s)$

$\vdots$

$\leq \mathbb{E}_{\pi'}(R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4} + \cdots \mid S_t = s) = v_{\pi'}(s)$

Уместо разматрања побољшавања политике у једном стању за једну акцију, можемо разматрати за сва стања и све могуће акције, у сваком стању бирајући најбољу акцију према $q_\pi(s,a)$. Другим речима, разматрамо нову похлепну политику π':

$$\pi'(s) = \underset{a}{\operatorname{argmax}}\, q_\pi(s,a) = \underset{a}{\operatorname{argmax}} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_\pi(s')\big)$$

где $\underset{a}{\operatorname{argmax}}$ означава акцију са највећом вредношћу $q_\pi(s,a)$. Политика је похлепна зато што узима најбољу акцију само на основу разматрања следећег корака, не гледајући даље у будућност. Међутим, π' задовољава услове Теореме побољшавања политике, па је боља или подједанко добра као и оригинална политика.

Процес конструисања овакве политике (похлепне у односу на функцију вредности стања за оригиналну политику) се зове побољшавање политике.

Нека је нова политика подједнако добра као и оригинална, али није боља (ни за једно стање). Тада је $v_\pi = v_{\pi'}$ и, на основу претходне једнакости,

$$v_{\pi'}(s) = \max_a \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_\pi(s')\big)$$

Ово је заправо Белманова једначина оптималности, па су и π и π' оптималне политике.

1.2.3. Итерација

После побољшавања политике следи рачунање функције вредности стања у односу на нову политику. Затим се нова политика побољшава и опет рачуна функција вредности. Кад политика више не може да се побољша, пронађена је оптимална политика. ([1], поглавља 4.3 – 4.5.)

Овакав метод има неколико мана чија је последица превише утрошеног времена да се дође до оптималне политике.

Прво побољшање је свакако заустављање итерације пре налажења оптималне политике кад је тренутна политика довољно добра. тј када је разлика функција вредности тренутне и претходне политике мања од унапред датог броја.

Велика мана је рачунање функција вредности за сва стања у сваком кораку. У случају великог броја стања, потребне су године за завршавање процеса.

Асинхроно динамичко програмирање, за разлику од претходно разматраних алгоритама, не рачуна функцију вредности за сва стања у сваком кораку, већ само за посећена стања за које је и могуће ажурирање функције вредности. Тиме се постиже следећи ефекат: Агент учи да доноси добре одлуке у стањима која се релативно често посећују, док се може понашати врло лоше у осталим стањима. На пример, претпоставимо да агент учи да игра шах. Шах има веома много различитих позиција. Да би агент био добар играч, довољно је да научи да игра добро у позицијама које се сусрећу приликом игре са добрим играчима. Остале позиције се сусрећу у игри са лошим играчима, а против њих није ни потребно одиграти најбоље потезе.

1.3. *Temporal-difference* учење

Динамичко програмирање и поред свих побољшања не показује добре резултате у пракси. Сетимо се три услова неопходних за налажење оптималне политике динамичким програмирањем: ([1], глава 6.)

- тачно знамо динамику окружења;

- имамо довољну компјутерску снагу;
- марковско својство.

Можемо рећи да тачно знамо динамику окружења, међутим, у пракси, то није тачно. Покер има веома велики број комбинација различитх карата на столу и у руци и рачунање свих могућих вероватноћа захтева превише времена. Дакле, не знамо динамику окружења. Такође немамо ни довољну компјутерску снагу. Вратимо се на итеративну формулу динамичког програмирања:

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)\big(r + v_k(s')\big)$$

Пошто ћемо пратити само детерминистичке политике – бира се само акција са највећом вредношћу, $\pi(a|s)$ је 1 за најбољу акцију и 0 за остале, па можемо обрисати спољну суму.

Како не можемо израчунати вероватноће $p(s', r \mid s, a)$, морамо се послужити Монте-Карло методама, тј. учењем искључиво из искуства без помоћи знања динамике окружења. Тада ће се вероватноће преласка одразити на учесталост стања у које ће процес долазити.

Формула се своди на

$$v_{k+1}(s) = r + v_k(s')$$

Пређимо на следећу нотацију:

$$v(s_k) = r + v(s_{k+1})$$

Како не знамо у које стање ће процес прећи, морамо користити аналогну формулу за парове (стање,акција):

$q(s_k, a_k) = r_{k+1} + q(s_{k+1}, a_{k+1})$, за све $a \in \mathcal{A}(s)$.

1.3.1. Алгоритам *Sarsa*

Temopral-difference (ТД) учење је револуционарни напредак у теорији учења условљавањем. Све у пракси коришћене методе су примене ТД учења на различите врсте проблема. Оно се, у основи, састоји у следећем: Како не знамо у које стање ће процес прећи после одабира акције a, агенту се мора дати време да прође кроз прелазе у различита стања и да осети последице свих исхода. ([1], поглавље 6.4.)

Зато се у горњу формулу, која се може приказати на следећи начин:

$$q(s_k, a_k) = r_{k+1} + q(s_{k+1}, a_{k+1}) = 0 * q(s_k, a_k) + 1 * \big(r_{k+1} + q(s_{k+1}, a_{k+1})\big)$$

додаје метапараметар:

$$q(s_k, a_k) = (1 - \alpha) * q(s_k, a_k) + \alpha * \big(r_{k+1} + q(s_{k+1}, a_{k+1})\big)$$
$$= q(s_k, a_k) + \alpha * \big(r_{k+1} + q(s_{k+1}, a_{k+1}) - q(s_k, a_k)\big)$$

тј. функција вредности акције се ажурира тако што јој се додаје:

$$\alpha * \big(r_{k+1} + q(s_{k+1}, a_{k+1}) - q(s_k, a_k)\big)$$

Алгоритам је добио име по низу $(s_k, a_k, r_{k+1}, s_{k+1}, a_{k+1})$.

Упоредимо алгоритам Sarsa са обичним ТД учењем. По обичном ТД учењу, вредност $q(s_k, a_k)$ се Монте-Карло методом добија на следећи начин:

$$q(s_k, a_k) = \sum_{i=1}^{n} \frac{r_i}{n} + q(s_{k+1}, a_{k+1})$$

Прва предност алгоритма *Sarsa* је што није потребно чувати податке о броју одабира сваке акције. При вишемесечном раду програма, ови бројеви лако могу да изађу из опсега целобројног типа података, што онда захтева додатне ресурсе – више меморије за чување бројева на други начин и више времена за обрату истих.

Друга и главна предност је следећа: Обично ТД учење бива „заробљено у прошлости“, тј. утицај нових података временом опада да би на крају постао занемарљив. Код алгоритма Sarsa, новији, а самим тим и тачнији, подаци имају већи утицај – њихов удео је увек α, док се утицај сваког конкретног одабира акције временом смањује, тј. застарева. То још значи да се агент прилагођава променама у окружењу, што је особина коју желимо да има.

2. Правила игре

2.1. Увод

Покер се игра са једним целим шпилом од 52 карте. Постоје 13 различитих бројева: 2, 3, 4, 5, 6, 7, 8, 9, 10, J (жандар), Q (дама), K (краљ) и A (ас или кец). Шпил има по 4 карте сваког броја, различитих знакова, тј. боја: ♠ (пик), ♣ (треф), ♦ (каро) и ♥ (херц). Цео шпил је приказан у следећој табели:

	2	3	4	5	6	7	8	9	10	жандар	дама	краљ	ас
пик	♠2	♠3	♠4	♠5	♠6	♠7	♠8	♠9	♠10	♠J	♠Q	♠K	♠A
треф	♣2	♣3	♣4	♣5	♣6	♣7	♣8	♣9	♣10	♣J	♣Q	♣K	♣A
каро	♦2	♦3	♦4	♦5	♦6	♦7	♦8	♦9	♦10	♦J	♦Q	♦K	♦A
херц	♥2	♥3	♥4	♥5	♥6	♥7	♥8	♥9	♥10	♥J	♥Q	♥K	♥A

Циљ игре је да се заради новац уз добру забаву. Данас постоји много различитих варијанти покера, а најпопуларнија је Тексас Холдем. Пореклом је из Тексаса, а холд 'ем означава да се користе своје карте (које се држе у руци) и заједничке карте (које су на столу). Термин долази од изазова да се карте задрже у руци (*to hold 'em (them)*), тј. да се остане у игри кроз неколико кругова улагања, уместо повлачења из игре (*fold 'em*).

Дефиниција: Рука је:

1. Један круг игре који почиње са обавезним улагањем, а завршава се освајањем уложеног новца од стране једног или више играча.
2. Најбољи избор од 5 карата.

Дефиниција: *Blind* је обавезан улог за једног или два играча на које је дошао ред. То је улагање „на слепо“, не виде се карте (зато што још нису подељене), па отуда и име.

Дефиниција: *Ante* је обавезан улог за све играче, такође пре дељења карата.

Постоје два начина игре:

1. Слободна игра: Свако може да се прикључи игри са колико год новца жели. Како би се обезбедили равноправни услови, ограничавају се највећи и најмањи дозвољен унос новца. Свако може у сваком тренутку да изађе из игре и однесе новац који му је остао. Најчешће се користе *small blind* и *big blind*, а у неким варијантама се користи *ante*.
2. Турнири: Сваки играч плаћа улазницу и од њих се формира наградни фонд. После сваког истека одређеног временског интервала или одређеног броја руку, *blind*-ови се повећавају, а после одређеног времена се додаје и *ante*, да би се ограничило трајање турнира.

2.2. Рука

Рука се састоји од наизменичног понављања улагања и дељења карата, а завршава се упоређивањем карата играча који су остали у игри и освајањем уложеног новца.

Основно правило сваког круга улагања је да сви играчи морају да уложе исти износ новца (једини изузетак је ако неко нема довољно новца, онда улаже све што има) или да се повуче из руке и врати карте делиоцу.

Графички приказ тока руке се налази на дијаграму 1.

2.2.1. Pre-flop

Игра почиње од играча који дели карте (engl. dealer). Испред њега стоји жетон који се после сваке руке помера код играча са његове леве стране. Игра се одвија у смеру казаљке на сату. Следећи играч улаже *small blind*, а следећи *big blind*. Они се обично зову тим именима. Затим се сваком играчу деле по две карте.

Према основном правилу улагања, играч има избор да уложи износ *big blind*, да се повуче из руке или да подигне улог.

Дефиниција: Играч прати улог (*call*) ако улаже исти износ новца колико и играч који је највише уложио.

Дефиниција: Играч враћа карте (*fold*) ако не жели да испрати улог и повлачи се из руке.

Дефиниција: Играч подиже улог (*raise*) ако уложи више новца него што је неопходно да би испратио постојећи улог.

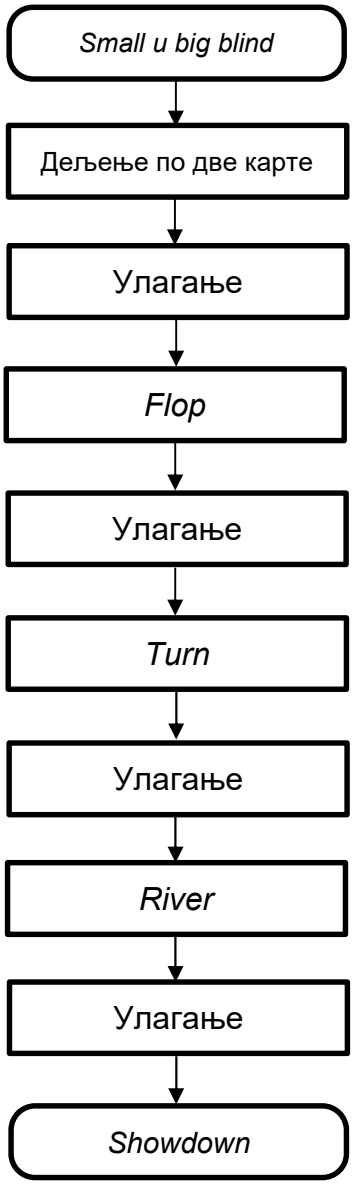
Најмањи дозвољен износ подизања улога је дуплирање улога, а највећи улагање свега[1].

Дефиниција: *All-in* је кад играч уложи сав новац.

У случају да играч нема довољно новца да испрати улог, има две опције: *all-in* или *fold*. Тада се укупан улог дели на два дела на следећи начин: Сав улог из претходних кругова улагања се ставља у први део. У тренутном кругу улагања, сви играчи који су остали у игри у први део стављају онолико новца колико је уложио играч који нема довољно, а остало у други део. Играч који је одиграо *all-in* највише може да освоји сав новац из првог дела.

Ако је само један играч остао у игри, тј. сви остали су вратили карте, он осваја сав уложен новац и почиње следећа рука.

Ако се у било ком кругу улагања деси да су сви играчи осим једног уложили сав новац, тј. нико више не може ништа да уложи, сви играчи отварају, тј. показују карте и делилац дели све преостале заједничке карте.



Дијаграм 1:
Ток руке

[1] То важи ако су правила неограниченог улагања.

2.2.2. Flop

Кад је завршен први круг улагања, делилац извлачи три карте и поставља их на сто тако да их сви виде. Затим иде следећи круг улагања.

На реду је први играч са леве стране делиоца који није вратио карте. Како нема улог да се прати он има следеће три опције: да не одигра ништа, да подигне улог или да врати карте. Трећа опција нема смисла јер се може бесплатно остати у игри, па заправо остају две опције.

Дефиниција: *Check* је кад играч не мора и не жели ништа да уложи.

Дефиниција: *Bet* је кад играч не мора ништа да уложи, а одлучи да уложи.

Најмањи износ улога је тада *big blind*, а највећи *all-in*[2]. После се улагање одвија исто као и пре флопа, тј. *raise*, *call* или *fold*.

2.2.3. *Turn* и *river*

После круга улагања после флопа, делилац извлачи једну карту и ставља је отворену поред претходне три. Улагање се одвија исто као и после флопа. Затим се вуче још једна карта и следи последњи круг улагања.

2.2.4. *Showdown*

Ако нико није победио до сада, играчи који су остали у игри показују своје карте и коју руку (по другом делу дефиниције) имају. Сваки играч има 7 карата, 5 отворених и 2 његове, и бира 5 тако да састави најбољу могућу руку. Могу се изабрати и свих 5 заједничких карата. Играч са најјачом руком осваја сав уложени новац. Ако више играча имају исту најјачу руку, они деле улог.

Дефиниција: Број на карти се зове и јачина.

Руке се по структрури деле на неколико категорија које се рангирају према њиховој вероватноћи. Руке из исте категорије се рангирају на следећи начин: свака рука се може јединствено представити низом дужине од 1 до 5 елемената. Онда се пореде редом по елементима низа, почев од првог. Ако су сви елементи низа исти, тада су свих 5 карата обе руке исте и оне су исте јачине.

Категорије су следеће:

- 5 „дасака“ или ништа (engl. *high card*) – ниједна од следећих руку
  ♠K ♦J ♥8 ♠6 ♣4
  Низ се састоји од сортираних јачина свих 5 карата.
- Пар – две карте истог броја и све остале различите карте
  ♠7 ♦7 ♥Q ♠6 ♣4
  Први елемент низа је јачина пара, а затим сортиране јачине остале три карте.
- Два пара – две карте са једним бројем, две са другим бројем и једна са трећим бројем
  ♥Q ♠Q ♠7 ♦7 ♣4
  Низ је: (јачи пар, слабији пар, пета карта).

[2] Исто.

- Трилинг – три карте истог броја и остале различите карте
  ♠7 ♦7 ♣7 ♥Q ♠8
  Низ је: (јачина трилинга, јача преостала карта, слабија преостала карта).
- Кента (engl. *straight*) – пет карата са бројевима у низу
  ♠9 ♦8 ♣7 ♥6 ♠5
  Низ има један елемент: најјачу карту.
- Боја (engl. *flush*) – пет карата истог знака
  Низ се састоји од сортираних јачина свих 5 карата.
  ♥A ♥J ♥10 ♥8 ♥3
- Фул (engl. *full house*) – три карте једног броја и две карте другог броја
  Низ је: (јачина трилинга, јачина пара).
  ♠7 ♦7 ♣7 ♥Q ♠Q
- Покер – четири карте истог броја
  ♠7 ♦7 ♣7 ♥7 ♠2
  Низ је: (јачина покера, пета карта).
- Кента у боји – пет карата са бројевима у низу истог знака
  ♥Q ♥J ♥10 ♥9 ♥8
  Низ има један елемент: најјачу карту.
- Флеш ројал (engl. *royal flush*) – кента у боји са картама од 10 до А.
  ♥A ♥K ♥Q ♥J ♥10
  Низ има један елемент: А.

Играч који је уложио сав новац и изгубио руку испада из турнира. Ако је слободна игра, може да изађе из игре или да унесе још новца.

2.3. Примери из програма

Напомена: У Јави се не могу једноставно исписати карташки знаци (пик, треф, каро и херц), тако да смо се одлучили за исписивање словима.

Пример 1

```
igrac 2       2030$
igrac 3        995$
igrac 4        985$
igrac 5        990$
igrac 6       1000$
igrac 7       1000$
igrac 8       1000$
igrac 9       1000$
igrac 10      1000$
igrac 11      1000$
igrac 5        985$ pik  Q   herc J   small blind  5
igrac 6        990$ herc 8   tref 9   big   blind  10
igrac 7       1000$ karo 4   tref 5   fold
```

```
igrac 8       1000$ pik  5   tref K   fold
igrac 9       1000$ tref 4   karo J   fold
igrac 10      1000$ pik  4   tref 8   fold
igrac 11      1000$ karo 3   pik  2   fold
igrac 2       2030$ herc 6   pik  7   call   10
igrac 3        995$ tref J   herc 7   fold
igrac 4        985$ herc 9   tref Q   fold
igrac 5        985$ pik  Q   herc J   raise  30
igrac 6        990$ herc 8   tref 9   fold
igrac 2       2020$ herc 6   pik  7   call   30

ukupan ulog: 70
 karo A   pik  9   herc 5
igrac 5        960$ pik  Q   herc J   45% check
igrac 2       2000$ herc 6   pik  7   35% check

ukupan ulog: 70
 karo A   pik  9   herc 5   herc 3
igrac 5        960$ pik  Q   herc J   34% check
igrac 2       2000$ herc 6   pik  7   28% check

ukupan ulog: 70
 karo A   pik  9   herc 5   herc 3   karo 7
igrac 5        960$ pik  Q   herc J   23% bet    35
igrac 2       2000$ herc 6   pik  7   60% call   35

ukupan ulog: 140

igrac 2 herc 6   pik  7   nosi 140$.
```

Пример 2

```
igrac 1        995$
igrac 2       1015$
igrac 3        985$
igrac 4        855$
igrac 5       1145$
igrac 6        985$
igrac 7       1000$
igrac 8       1005$
igrac 9       1000$
igrac 10      1015$
igrac 8       1000$ tref 6   karo A   small blind  5
igrac 9        990$ pik  7   pik  8   big   blind  10
igrac 10      1015$ tref A   tref Q   raise  550
igrac 1        995$ tref 8   tref 10  fold
igrac 2       1015$ karo 3   herc 10  fold
igrac 3        985$ tref 4   karo 4   fold
igrac 4        855$ pik  3   karo J   fold
```

```
igrac 5       1145$ herc J   herc A   call   550
igrac 6        985$ tref 3   pik  2   fold
igrac 7       1000$ herc 8   herc 3   fold
igrac 8       1000$ tref 6   karo A   fold
igrac 9        990$ pik  7   pik  8   fold

ukupan ulog: 1115
 karo 9   tref 7   karo 5
igrac 10       465$ tref A   tref Q   48% check
igrac 5        595$ herc J   herc A   46% bet    465
igrac 10       465$ tref A   tref Q   48% all-in 465

ukupan ulog: 2045
 karo 9   tref 7   karo 5   pik  K

ukupan ulog: 2045
 karo 9   tref 7   karo 5   pik  K   pik  6

ukupan ulog: 2045

igrac 10 tref A   tref Q   nosi 2045$.
```

Пример 3

```
igrac 1        995$
igrac 2       1015$
igrac 3        985$
igrac 4        855$
igrac 5        130$
igrac 6        985$
igrac 7       1000$
igrac 8       1000$
igrac 9        990$
igrac 10      2045$
igrac 9        985$ karo 5   tref A   small blind  5
igrac 10      2035$ herc 8   karo Q   big   blind  10
igrac 1        995$ pik  2   karo 9   fold
igrac 2       1015$ karo 7   pik  9   fold
igrac 3        985$ tref 7   pik  5   fold
igrac 4        855$ tref 8   herc 5   fold
igrac 5        130$ herc A   pik  8   fold
igrac 6        985$ tref 6   herc K   fold
igrac 7       1000$ pik  K   karo K   raise  130
igrac 8       1000$ karo J   herc 3   fold
igrac 9        985$ karo 5   tref A   fold
igrac 10      2035$ herc 8   karo Q   call   130

ukupan ulog: 265
 pik  J   herc 7   pik  Q
```

```
igrac 10      1915$ herc 8   karo Q   48% check
igrac 7        870$ pik  K   karo K   85% check

ukupan ulog: 265
 pik  J   herc 7   pik  Q   karo 2
igrac 10      1915$ herc 8   karo Q   83% check
igrac 7        870$ pik  K   karo K   87% check

ukupan ulog: 265
 pik  J   herc 7   pik  Q   karo 2   pik  A
igrac 10      1915$ herc 8   karo Q   72% check
igrac 7        870$ pik  K   karo K   75% bet    795
igrac 10      1915$ herc 8   karo Q   72% call   795

ukupan ulog: 1855

igrac 7 pik  K   karo K   nosi 1855$.
```

Пример 4

```
RUKA 68
igrac 67       190$
igrac 71       175$
igrac 72       180$
igrac 77       178$
igrac 78       663$
igrac 80       985$
igrac 81      3985$
igrac 84      1000$
igrac 85      1000$
igrac 86      1000$
igrac 84       995$ karo 8   pik  5   small blind  5
igrac 85       990$ herc Q   karo 10  big   blind  10
igrac 86      1000$ tref 9   tref Q   call   10
igrac 67       190$ karo 7   karo 2   fold
igrac 71       175$ pik  2   pik  6   fold
igrac 72       180$ tref J   herc 5   fold
igrac 77       178$ tref 3   pik  9   fold
igrac 78       663$ pik  K   tref 8   call   10
igrac 80       985$ herc A   karo A   call   10
igrac 81      3985$ karo Q   pik  A   raise  340
igrac 84       995$ karo 8   pik  5   fold
igrac 85       990$ herc Q   karo 10  call   340
igrac 86       990$ tref 9   tref Q   call   340
igrac 78       653$ pik  K   tref 8   call   340
igrac 80       975$ herc A   karo A   call   340
```

```
ukupan ulog: 1705
 herc 7   herc 8   herc 10
igrac 85       660$ herc Q   karo 10  46% check
igrac 86       660$ tref 9   tref Q   18% check
igrac 78       323$ pik  K   tref 8   16% check
igrac 80       645$ herc A   karo A   57% check
igrac 81      3645$ karo Q   pik  A   92% check

ukupan ulog: 1705
 herc 7   herc 8   herc 10  tref 6
igrac 85       660$ herc Q   karo 10  29% check
igrac 86       660$ tref 9   tref Q   51% check
igrac 78       323$ pik  K   tref 8    4% check
igrac 80       645$ herc A   karo A   36% check
igrac 81      3645$ karo Q   pik  A   95% bet    31
igrac 85       660$ herc Q   karo 10  29% call   31
igrac 86       660$ tref 9   tref Q   51% call   31
igrac 78       323$ pik  K   tref 8    4% call   31
igrac 80       645$ herc A   karo A   36% call   31

ukupan ulog: 1860
 herc 7   herc 8   herc 10  tref 6   pik  10
igrac 85       629$ herc Q   karo 10  37% check
igrac 86       629$ tref 9   tref Q   58% check
igrac 78       292$ pik  K   tref 8    4% check
igrac 80       614$ herc A   karo A   27% check
igrac 81      3614$ karo Q   pik  A   96% bet    629
igrac 85       629$ herc Q   karo 10  37% all-in 629
igrac 86       629$ tref 9   tref Q   58% all-in 629
igrac 78       292$ pik  K   tref 8    4% fold
igrac 80       614$ herc A   karo A   27% all-in 614
```

3. Стратегија

Циљ игре је да се заради што више новца у просеку, тј. на дуже време. Новац се може зарадити на два начина: показивањем најјаче руке у *showdown*-у или останком у игри кад сви остали играчи врате карте. У оба случаја, циљ се постиже улагањем новца, тако да је стратегија у покеру одговор на питање да ли и колико новца уложити у различитим случајевима.

Тема овог рада није тражење оптималних стратегија и математичко извођење истих, већ постоје истраживања и материјали на ту тему. ([3]) Потребно је анализирати концепте на којима се стратегија заснива да би се успешно направио модел за машинско учење.

3.1. Пре флопа

Налазимо се у наизглед једноставној ситуацији – имамо две карте у руци и, знајући све могуће комбинације, знамо њихову јачину. Међутим, на крају руке ћемо имати 7 карата на основу којих ће се знати ко ће однети уложен новац. Дакле, испоставља се да знамо врло мало и да је доношење одлуке у овом кругу најтеже.

Почнимо од два начина да зарадимо новац: показивањем најјаче руке или избацивањем осталих играча из игре. Избацивање осталих играча се постиже блефирањем, међутим, оно не може да буде успешно ако се често покушава, па остаје показивање најјаче руке.

Прва идеја је да једноставно испратимо улог и видимо флоп, а затим да улажемо кад видимо какве су наше и противничке могућности. Најслабија и највероватнија рука је пар. Претпоставимо да имамо различите карте у руци. Вероватноћа да саставимо најмање пар на флопу је око 50%. Пошто један пар често није довољно јака рука да бисмо очекивали победу, а пар може да буде и од 3 заједничке карте, верованоћа да на флопу имамо два пара или јаче је око 7.62%. Дакле, тек на сваком тринаестом флопу [3] ћемо имати довољно јаку руку да бисмо играли на победу. У осталих 12 ћемо изаћи из руке чим неко подигне улог.

Долазимо до закључка да не треба увек ући у игру, него треба играти одређене почетне руке (тј. две карте). Да бисмо одредили које почетне руке да играмо, размотримо све могуће руке које можемо имати на крају и како лакше доћи до њих:

- Кента у боји и покер долазе веома ретко.
- Фул ћемо вероватније добити ако у руци имамо пар.
- Боју ћемо вероватније добити ако у руци имамо карте исте боје.
- Кенту ћемо највероватније добити ако у руци имамо карте у низу, од 4 и 5 до 10 и J. Она се може циљати и ако има „размак“ не већи од 3 броја.
- Трилинг ћемо вероватније добити ако у руци имамо пар.
- Да бисмо победили са паром или чак и ни са чим, потребно је имати јаке карте у руци.

[3] Ово је врло непрецизно разматрање, али довољно да се види да је приступ погрешан.

Дакле, две карте имају потенцијал ако су истог броја, исте боје, сличних бројева или јаке.

Да ли ући у руку или не такође зависи и од броја играча у игри. На пример, ако играмо против два играча, довољно је да само једна карта буде бар дама, а друга било шта, док, ако играмо против 9 играча, улазимо у игру само са најбољим картама – у низу исте боје или обе бар жандар.

Пошто знамо са којим картама да улазимо у руку, следеће питање је колико уложити. Наравно, то опет зависи од броја играча и других фактора, а идеја је следећа: Ако, на пример, имамо A и K у руци, и ако се на флопу појави други ас, ниједан други играч који има аса неће моћи да победи на основу пара кечева. То даје идеју да се подигне улог пре флопа и, кад саставимо неку руку на флопу, освојићемо више новца. Са друге стране, ако имамо 10 и J, онда је боље не дизати улог и чекати да имамо нешто вредно после флопа.

Закључак: са боље две карте се подиже улог, односно прате већи улози пре флопа, иначе не. Ако почетна рука не спада ни у једну од 4 наведене категорије, не улазимо у руку.

За неколико најјачих руку: A A, K K, Q Q и A K, најбоље је уложити што више новца пре флопа због врло високе вероватноће победе против осталих руку. То ће уједно избацити играче са слабије две карте које би можда могле да саставе јачу руку на флопу.

3.2. После флопа

Кад знамо 5 од 7 карата које ћемо имати, можемо да одлучимо како ћемо играти до краја руке. Основни начин зарађивања је оправдано улагање новца, тј. ако верујемо да ће наша рука бити најјача ако дође до *showdown*-а. Блефирање је помоћни начин ако се ситуација промени после извлачења још заједничких карата или, уопштено, ако се укаже прилика, зато што често блефирање доводи до неповерења противника, који ће престати да враћају карте и узимати нам новац.

Разликујемо следеће случајеве квалитета наше руке:

- врло јака рука
- рањива јака рука
- *draw* – чекање јаке руке
- слаба рука
- безвредна рука

Најзначајније руке које се чекају су боја и кента зато што, када их дочекамо, углавном имамо врло јаку руку. Такође, обично немамо никакву другу руку или пар, па је велика разлика ако их дочекамо или не.

Наша игра такође зависи и од 3 карте флопа. Флоп може бити влажан (engl. wet) и сув (engl. dry). Флоп је влажан ако су карте такве да омогућавају противницима да чекају кенту или боју, иначе је сув. На пример, флоп ♠A, ♥8 и ♦2 је сув, док је ♣10, ♣9 и ♣8 врло влажан. Флоп може да буде и мање влажан, на пример: ♠A, ♥8 и ♦7; ♠A, ♥8 и ♥2 или ♠A, ♥8 и ♥7.

3.2.1. Врло јака рука

Најпожељнији случај је да после флопа имамо најјачу или вероватно најјачу руку, тј. противнику су потребне обе његове карте да би саставио бољу руку. Јачу руку од фула нећемо разматрати због веома мале вероватноће (0,2% од свих 7 карата).

У следећим примерима имамо најјачу руку – не постоји начин да неко има бољу руку (engl. nut-hand):

10 Q [4]	10 10 Q Q 2 [5]
♠A ♦7	♠K ♠8 ♠2 ♥7 ♠3
♠9 ♦10	♣J ♣8 ♠2 ♥7 ♠3
♠Q ♦Q	♣Q ♣8 ♠2 ♥7 ♠3

У следећим примерима имамо вероватно најјачу руку:

A Q	10 10 Q Q K

Можемо да изгубимо ако противник има 10 10, K K или K Q.

♠K ♠3	♠Q ♠8 ♠2 ♥8 ♥7

Можемо да изгубимо ако противник има фул или ♠A и још једну карту боје ♠.

♠2 ♦10	♣J ♣8 ♠9 ♥7 ♠3

Можемо да изгубимо ако противник има 10 и Q.

♠7 ♦7	♣J ♠J ♣8 ♥7 ♠3

Можемо да изгубимо ако противник има J J, J 8, J 7, J 3 или 8 8.

♠8 ♦A	♣A ♠J ♣8 ♥7 ♠3

Можемо да изгубимо ако противник има A A, J J, 8 8, 7 7, 3 3 или A J.

Како је у свим овим примерима вероватноћа пораза изузетно мала, можемо да претпоставимо да ћемо сигурно победити, па да тако и играмо.

Желимо да зарадимо што више новца, тако да улози треба да буду умерени, да би остали играчи испратили и повећали нам зараду. Ако неко подигне улог, у најгорем случају има исто што и ми, па нема разлога за страх од губитка новца.

Ако играмо против играча који често улажу, или су заједничке карте такве да је врло могућа јака рука, али и даље слабија од наше:

7 7	J J 8 7 3
♣Q ♦7	♠Q ♠8 ♠2 ♥7 ♠7

добра идеја је да улажемо врло мало или ништа, да бисмо испровоцирали противнике да улажу и тако освојимо још више новца.

A 10	10 10 Q Q K

У овом случају се игра пажљивије и са мањим улозима, зато што је противнику потребно да једна карта буде Q. Како су у шпилу преостале само две даме, није велика вероватноћа да изгубимо, нарочито ако играмо против једног или два играча.

♠K ♦7	♠Q ♠8 ♠2 ♥7 ♠3

Исто и овде, ако противник има ♠A.

---

[4] У свим оваквим примерима, две леве карте су карте које имамо у руци, а остале карте су заједничке.

[5] Боја карата се не наводи јер не утиче на руку, па би смањивала прегледност.

Ако неко подигне улог, не мора да значи да има бољу руку, може да буде и блеф, тако да треба добро проценити на основу претходног понашања тог играча.

3.2.2. Рањива јака рука

У претходном поглављу су навођени примери свих јаких руку од два пара до фула. Већина тих руку могу да буду рањиве:

♠4 ♠3	♠Q ♠8 ♠2
♠7 ♦10	♣J ♣8 ♠9
♠9 ♦10	♣J ♣8 ♠2
♠Q ♦Q	♣Q ♣8 ♠2
♠9 ♦A	♣A ♠J ♣9

а могу да буду и већ побеђене:

♠2 ♦10	♣J ♣8 ♠9 ♣7
♠9 ♦10	♣J ♣8 ♣2
♠Q ♦Q	♣Q ♣8 ♣2
♠10 ♦A	♣A ♠J ♣10

У другом случају се игра исто као и у горе наведеном примеру (A 10).

У првом случају тренутно имамо најјачу или вероватно најјачу руку, али прети озбиљна опасност да на турну или риверу буде извучена карта која ће омогућити противницима да имају кенту или боју. Тада је потребно, пре него што се то деси, натерати их да се повуку из руке високим улозима. Узмимо пример:

♠Q ♦Q	♣Q ♣8 ♠2

Претпоставимо да бар један играч има две карте боје ♣. Укупно их има 13, па остаје 9. Укупно је остало још 45 карата које не видимо. Вероватноћа да нека од следеће две карте буде баш те боје је 0.36, мало више од трећине. Ако после флопа уложимо онолико новца колико је до сад на столу и опет колико буде на столу после турна, противнику неће бити исплативо да чека боју. Ако ипак прати улог, зарадићемо у просеку ако чека боју, а ако има нешто друго, наша рука ће бити јача.

Ово се зове стратегија заштите рањиве руке.

3.2.3. *Draw* – чекање јаке руке

Ово је обрнути случај од горе наведеног, кад ми чекамо кенту или боју. При суочавању са одлуком да ли да пратимо улог или не, потребно је проценити вероватноћу и израчунати колики ће да буде удео новца кога будемо уложили у укупном новом улогу.

Како је тешко брзо израчунати тачну вероватноћу, важи следеће: Ако ћемо видети још само једну карту пре следећег улога или краја игре, онда се број карата које нам омогућују да саставимо кенту или боју (engl. outs) множи са 2/100. Ако ћемо пак видети обе карте, онда се множи са 4/100. Међутим, после турна, ако још није био, очекујемо још један улог. У супротном, ми улажемо и терамо противника да врати карте. Зато можемо увек множити са 4. Ситуација је још повољнија ако још неко прати улог, моћи ћемо да освојимо више новца за исту цену.

Примери:

♠9 ♦10 ♣J ♥7 ♠2

Чекамо једну од четири карте броја 8 – вероватноћа је око 0,16.

♠9 ♦10 ♣J ♥8 ♠2

Четири 7 и четири Q – вероватноћа је око 0,32.

♠4 ♠3 ♠Q ♠8 ♥2

9 карата боје ♠ – вероватноћа је око 0,36 (на пример, тачна вероватноћа у овом случају је 0,363636).

♣9 ♣10 ♣J ♣8 ♠2

9 карата боје ♣ и још три 7 и 3 Q, укупно 15 – вероватноћа је око 0,6. Ово је веома добра ситуација, јер се исплати да пратимо и највиши улог.

При бројању карата које чекамо не можемо увек да рачунамо све карте са којима ћемо добити жељену руку, на пример:

♠2 ♦8 ♣J ♥10 ♠9

Фали нам 7 или Q за кенту. Међутим, ако дође дама, неко са краљем ће имати јачу кенту, тако да, заправо, чекамо само седмицу, па је вероватноћа око 0,08.

♠9 ♦10 ♠J ♥8 ♠2

Чекамо четири седмице и четири даме. Међутим, ако дође ♠, неко може да има боју. У овом случају, пошто је за ту боју потребно да противник има обе карте ♠, рачунамо као пола карте које чекамо. Дакле, три и по седмице и три и по даме, па је вероватноћа око 0,14.

У случају да нико није ништа уложио, ако је у игри неколико играча, добра идеја је да подигнемо улог. Како немамо ниједну руку, то је блеф. Али, ако неко испрати, можемо се и даље надати кенти или боји, тако да је заправо полу-блеф.

Ако је у игри више играча, онда блеф уопштено није добра идеја, па ни ми не улажемо ништа и чекамо да видимо следећу карту.

3.2.4. Слаба рука

Слаба рука је пар. Слабе руке су такође оне које су слабије од најјаче руке коју противник може лако да има, на пример:

A K 10 10 Q Q K

имамо два пара, а противник може да има фул.

♠9 ♦10 ♣J ♣8 ♣2 ♥7 ♣3

кента – боја

♠Q ♦Q ♣Q ♣8 ♠9 ♥J ♠3

трилинг – кента

A K 10 10 Q 3 K

два пара – трилинг

Како је пар најечшћа рука која има неку снагу (за разлику од 5 „дасака“), посветићемо му више времена. Пар може бити:

- јачи од најјаче карте на столу (engl. *overpair*):
  K K 10 Q 3
- јак (engl. *top pair*) – састављен са најјачом картом на столу:
  K Q 10 Q 3
- средњи (engl. *middle pair*) – јачина пара је мања од најјаче карте на столу:
  K 10 10 Q 3
  J J 10 Q 3
- слаб (engl. *low pair*) – две или више карата на столу су јаче од јачине пара:
  K 3 10 Q 3
  8 8 10 Q 3
  2 2 10 Q 3

Јак пар или overpair се могу рачунати и у јаке рањиве руке, али се не рачунају зато што су било која два пара јача, а немогуће је видети има ли их неко само на основу карата на столу. Пар је рука највише подложна разним факторима, пре свега броју играча у игри и јачини неупарене карте. Наиме, често се дешава да више играча има исти пар. Онда неупарена карта у руци пресуђује ко односи новац. Зато се та карта зове и kicker. Под одређеним условима, да би јак пар био вредан улагања, неупарена карта мора бити такође јака, иначе га сврставамо у средњи пар.

Вероватноћа да један играч после флопа има два пара или јаче је око 0,076.[6] Међутим вероватноћа после ривера је око 0,39. За више играча вероватноћа је знатно већа. Ово нам говори да и јак пар мора да се штити не само од кенте и боје, већ и од два пара (у случају већег броја играча у игри), тако да је потребно постављати високе улоге да би се они избацили из игре пре него што им дође нека јача рука.

И средњи пар може бити релативно јака рука ако је још један играч у игри. Иначе, игра се пасивно и чека боља рука. Прате се само мали улози, осим ако имамо основане сумње да улагач блефира. Исто се игра и са слабим паром.

3.2.5. Безвредна рука

Најгори, а уједно и најчешћи случај на флопу. За разлику од средњег и малог пара, прати се само најмањи могући улог (у износу *big blind*-а). Циљ је да се дође до ривера и онда, ако нико други не уложи, уложимо малу количину новца, али већу од *big blind*-а. Други играчи обично имају слаб пар или 5 „дасака“, па неће пратити улог.

Претпоставимо да смо подигли улог пре флопа. Сетимо се, то радимо са малим бројем јаких почетних руку. Затим на флопу подижемо улог ако нико други већ није. Противници знају да имамо јаке карте, па нас неће пратити ако немају бар средњи или јак пар. Међутим, то се не ради ако је више играча у игри јер је вероватноћа да неко има бар средњи или јак пар довољно велика.

[6] Опет, тачна вероватноћа није битна, већ начин играња.

4. Формирање модела

Познавање основне стратегије описане у претходном поглављу је неопходно да би се направио добар модел, тј. да би се препознали сви важни атрибути. Свакако, најважнији атрибути су карте у руци и на столу, ако их има. Остали атрибути су:

2. <u>укупан број играча</u>
   Не треба улазити у игру пре флопа ако немамо довољно добре карте. Што више играча игра, више њих ће имати довољно добре карте.
3. <u>број играча који су подигли улог</u>
   Пре флопа:
   Играч који подиже улог најчешће има јаке карте. Ако још један играч подигне улог, онда он има врло јаке карте, што нам смањује вероватноћу победе у руци.
   После флопа:
   Прво улагање може бити блеф или заштита рањиве руке. Међутим, подизање улога од стране другог играча значи да он има врло јаку руку. Тада се улог прати само ако смо спреми да уложимо сав новац.
4. <u>број играча који су тренутно у игри</u>
   Слично као и претходно. Важно је користити оба атрибута зато што, ако су, на пример, 5 играча у игри, није исто да ли је на почетку било 5 или 10 играча. У другом случају 5 играча који су остали у игри имају боље карте и већа је вероватноћа да изгубимо.
5. <u>позиција</u>
   Означава колико је играча који су на реду после нас. Дели се на *blind*, рану, средњу и касну. Што касније дођемо на ред, имамо више информација о одлукама осталих играча и можемо да донесемо исправнију одлуку.
6. <u>однос вредности укупног уложеног новца и нашег неуложеног новца</u>
   Ако смо до сада уложили врло мало новца, немамо велики интерес да победимо у руци. Међутим, ако је на пример, укупан уложени новац три пута већи од новца који нам је остао, вреди борити се за руку и лакше је одлучити се за улагање још новца као и за блефирање.

У следећој табели су дати домени наведених атрибута, осим карата:

атрибут	домен
укупан број играча	{2, 3, 4, 5, 6, 7, 8, 9, 10}
број играча у игри	{2, 3, 4, 5, 6, 7, 8, 9, 10}
број играча који су подигли улог	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
позиција[7]	{-2, -1, 0, 1, 2, 3, 4, 5, 6, 7}
укупан улог/неуложен новац	$\mathbb{R}$

Број играча је ограничен на 10 зато што се у пракси не сређу столови са више играча.

Карте се представљају на два начина. Пре флопа, све што знамо су две карте, а како је укупан број комбинација те две карте (различите боје су еквивалентне) свега 169, најбоље

[7] Позиције -2 и -1 су позиције играча који улажу small и big blind. Они пре флопа играју последњи, а после флопа први.

решење је да се сваки случај третира као посебно стање. Међутим, кад се узму у обизир остали атрибути и кардиналност њихових домена, потребно је груписати сличне почетне руке у једно стање, што је објашњено у следећем поглављу.

После флопа, број различитих комбинација карата постаје изузетно велики, а дефинисање стања преко различитих карата је врло неповољно за поређење различитих стања, што је неопходно за машинско учење над великим простором стања. Такође, није довољно да анализирамо карте помоћу којих састављамо руку, већ морамо да анализирамо и све могућности других играча. Да би се тај проблем решио, рачуна се вероватноћа победе са датим картама и користи уместо њих.

Са оволиким скупом стања, а посебно са непрекидним скуповима вероватноћа и односа улога и уложеног новца, треба прећи са класичног учења условљавањем (*reinforcement learning*) на функционалну апроксимацију уз помоћ густих неуронских мрежа, што је и покушано. Међутим, у овом случају и са ограниченим рачунарским капацитетом, то није показало успех. Наиме, излаз неуронске мреже је за све вредности параметара увек исти број.

Први разлог је пораз са великим вероватноћама за победу. Занемаримо случајеве када играч са малом вероватноћом има среће да на риверу буде извучена баш она карта која му треба да победи. На пример, на столу су следеће карте:

10 10 J 3 9

Један играч има A 10, други K Q, трећи 3 3, а четврти 10 9. Сви играчи имају јаке руке и велике вероватноће да победе, али само један играч побеђује, а остали губе.

Са друге стране, претпоставимо да је за дату ситуацију са 4 играча у игри просечан профит 1. То може да значи да, од четири руке, три пута изгубимо улог x, а једном добијемо улог 4x. Неуронска мрежа види три пораза и само једну победу при великим вероватноћама.

Решење је коришћење једноставнијег начина машинског учења – помоћу матрице вредности стања.

4.1. Упрошћавање модела

Непрекидни скупови се морају дискретизовати.

Вероватноћа као најважнији атрибут се мора најдетаљније приказати, тј. да има знатно више стања од осталих атрибута. Изабране тачке интервала [0,1] су $\mathbb{Z}_{100}$/100, па су границе различитих стања (2*$\mathbb{Z}_{100}$+1)/200.

За однос укупног улога и неуложеног новца се користи логаритамска скала. Пожељно је задржати целе бројеве као границе. Као први начин се намеће функција 2^n. Међутим, размаци између граница су превелики и има премало стања: {1, 2, 4, 8, 16, 32}. Са Фибоначијевим низом (чије коришћење је идеја аутора рада) уз додатак броја 1,5 нема тог проблема: {1, 1.5, 2, 3, 5, 8, 13, 21}. Како укупан улог може бити и већи и мањи од неуложеног новца, потребно је додати и границе у интервалу [0,1], па је коначна подела: $\left\{\frac{1}{21}, \frac{1}{13}, \frac{1}{8}, \frac{1}{5}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1{,}5}, 1, 1.5, 2, 3, 5, 8, 13, 21\right\}$.

Сада је укупан број стања производ кардиналности домена свих атрибута: 3*101*15*10*10*9*9 = 36,814,500. Број стања пре флопа је нешто мањи, зато што однос укупног улога и неуложеног новца није релевантан: 169*10*10*9*9 = 1,368,900. У оба случаја,

број стања је изузетно велики и потребно је смањити га колико је могуће, а да квалитет модела остане сличан.

У следећој табели су дати упрошћени домени атрибута, односно границе интервала дискретизације непрекидних домена:

атрибут	домен	скуп стања атрибута[8]
вероватноћа	$\mathbb{R}$	{0.005, 0.015, 0.025, ..., 0.995}
укупан број играча	{2, 3, 4, 5, 6, 7, 8, 9, 10}	{2, 3, 4 и 5, 6 и 7, 8+}
број играча у игри	{2, 3, 4, 5, 6, 7, 8, 9, 10}	{2 и 3, 4 и 5, 6+}
број играча који су подигли улог	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}	{0, 1, 2+}
позиција	{-2, -1, 0, 1, 2, 3, 4, 5, 6, 7}	{-2 и -1, 0, 2 и 3, 4 и 5, 6+}
укупан улог/неуложен новац	$\mathbb{R}$	$\left\{\frac{1}{21}, \frac{1}{13}, \ldots, 13, 21\right\}$

И атрибут почетна рука се може упростити на основу стратегије улагања пре флопа. Сетимо се, прве две карте имају потенцијал ако су истог броја, исте боје, сличних бројева или јаке. Стања су следећа:

A A, K K, Q Q, J J, 10 10, 8 8 и 9 9, 6 6 и 7 7, 2 2 – 5 5, A K, A Q, A J, A 10, A 8-9, A 6-7, A 2-5, K Q, K J, K 10, K 8-9, K 6-7, K 2-5, Q J, Q 10, Q 8-9, Q 6-7, Q 2-5, 9 10 и 10 J, 8 10 и 9 J, 7 10 и 8 J, 10-J и све остало, 4 5 – 8 9, 3 5 – 7 9, 2 5 – 6 9, 9 и све остало.

Стања су исто груписана за карте исте и различите боје, осим парова.

Ако је најјача карта мања од 5, рачуна се као да је 5, зато што није могућа кента са највећом картом мањом од 5.

Графички приказ (за обе карте веће од 10 су сва стања различита):

<table>
<tr><td></td><td>10</td><td colspan="2">9</td><td>8</td><td colspan="2">7</td><td>6</td><td colspan="2">5</td><td>4</td><td>3</td><td>2</td></tr>
<tr><td>A</td><td>A 10</td><td colspan="3">A 8-9</td><td colspan="3">A 6-7</td><td colspan="5">A 2-5</td></tr>
<tr><td>K</td><td>K 10</td><td colspan="3">K 8-9</td><td colspan="3">K 6-7</td><td colspan="5">K 2-5</td></tr>
<tr><td>Q</td><td>Q 10</td><td colspan="3">Q 8-9</td><td colspan="3">Q 6-7</td><td colspan="5">Q 2-5</td></tr>
<tr><td>J</td><td>9 10 и 10 J</td><td colspan="2">8 10 и 9 J</td><td>7 10 и 8 J</td><td colspan="6" rowspan="2">10-J и било која преостала карта</td></tr>
<tr><td>10</td><td>10 10</td><td colspan="2">9 10 и 10 J</td><td>8 10 и 9 J</td><td>7 10 и 8 J</td></tr>
<tr><td>9</td><td rowspan="8"></td><td colspan="2">88-99</td><td>45-89</td><td>35-79</td><td>25-69</td><td colspan="5" rowspan="2">6-9 и било која преостала карта</td></tr>
<tr><td>8</td><td rowspan="7"></td><td>88-99</td><td>45-89</td><td>35-79</td><td>25-69</td></tr>
<tr><td>7</td><td rowspan="6"></td><td>66-77</td><td>45-89</td><td>35-79</td><td>25-69</td><td colspan="2"></td></tr>
<tr><td>6</td><td>66-77</td><td>45-89</td><td>35-79</td><td>25-69</td><td></td></tr>
<tr><td>5</td><td rowspan="4"></td><td>22-55</td><td>45-89</td><td>35-79</td><td>25-69</td></tr>
<tr><td>4</td><td>22-55</td><td>35-79</td><td>25-69</td></tr>
<tr><td>3</td><td rowspan="2"></td><td>22-55</td><td>25-69</td></tr>
<tr><td>2</td><td>22-55</td></tr>
</table>

Сада је укупан број стања пре флопа 60*5*4*3*3 = 10800, а после флопа по 101*15*5*5*4*3 = 454500 за флоп, турн и ривер.

4.2. Акције

[8] У случају непрекидног домена, дати скуп представља границе интервала.

Акције су, како је описано у правилима, *fold*, *check*, *bet* и *raise*. Пошто нема ограничења износа улога, играч сам бира колико ће новца да уложи. У пракси, пре флопа се износ улога везује за *big blind*, а после за укупан улог, обично 1, 3/4, 2/3 и 1/2. И овде је добра идеја користити Фибоначијев низ, из разлога који је наведен у претходном поглављу.

Могући улози пре флопа су *big blind* * {1, 2, 3, 5, 8, ...}, а после флопа укупан улог * $\left\{\dots, \frac{1}{8}, \frac{1}{5}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1{,}5}, 1, 1.5, 2, 3, 5, 8, \dots\right\}$.

Поставља се питање где поставити границу првог и последњег интервала, који ће да убухвате и граничне случајеве: улагање износа 0, што су *check* и *fold*, и улагње свега, тј. *all-in*.

Већина казина у пракси нуди играчима да унесу у игру новца у износу од 10 до 100 *big blind*-ова. За програм је изабран горњи гранични случај. Такође, кад играч заради 4 пута више новца него што је унео, тј. 500 * *big blind*, узима сав зарађени новац и напушта игру. Да не бисмо оптерећивали програм екстремним случајевима, претпоставимо да други најјачи играч неће имати више од половине тога, тј. 250 * *big blind*, тако да је то највише што најјачи играч може да освоји. Ако се не улаже сав новац, највећи улог који има смисла је половина. То даје следећи скуп акција: {1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, *all-in*}.

Ако играју два играча и обојица уложе најмање могуће, тј. *big blind*, укупно је уложено 2 * *big blind*, а највише је остало 249 * *big blind*, што је 124,5 * укупан улог. Највећи улог који има смисла је 1/3 * преостали новац да би у следећем кругу улагања било могуће да се улагањем свог преосталог новца противник натера на повлачење из руке.

Ако имамо веома јаку руку, желимо да улажемо мале количине новца. Опет, да не бисмо оптерећивали програм екстреминим случајевима (укупан улог може да буде већи и од 1000 * *big blind*), поставимо доњу границу на 1/144. То даје следећи скуп акција: $\left\{0, \frac{1}{144}, \frac{1}{89}, \frac{1}{55}, \frac{1}{34}, \frac{1}{21}, \frac{1}{13}, \frac{1}{8}, \frac{1}{5}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1{,}5}, 1, 1.5, 2, 3, 5, 8, 13, 21, 34, all-in\right\}$.

5. Опис програма

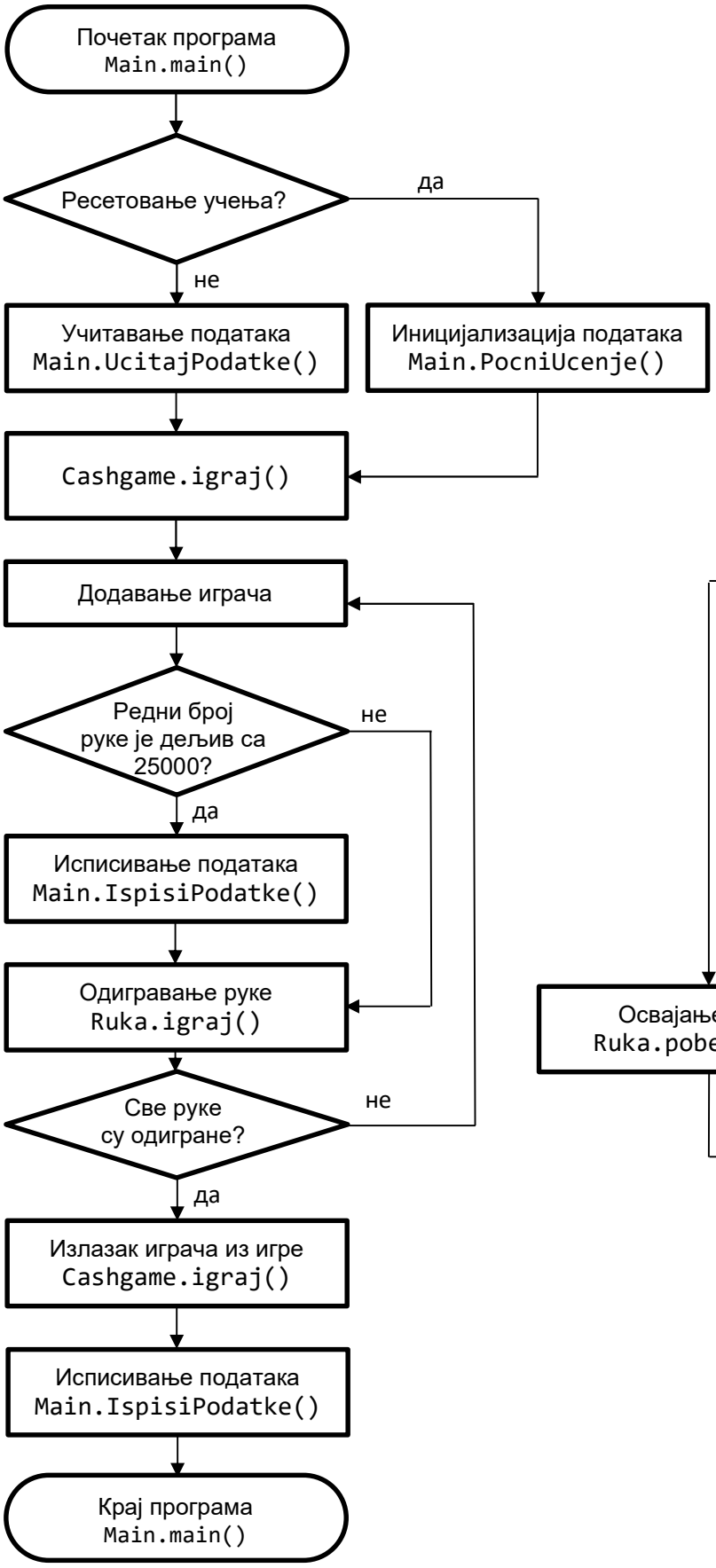
Програм ће радити по следећем моделу:

- Обавезни улози су *small* и *big blind* и они су 5$ и 10$.
- Играју укупно 10 играча. Свако почиње са 1000$. Игра се слободна игра (engl. *cashgame*), у којој свако има право да уђе и изађе из игре кад год жели.
- Кад играч остане без новца или сакупи 5000$, напушта игру, а улази следећи играч.

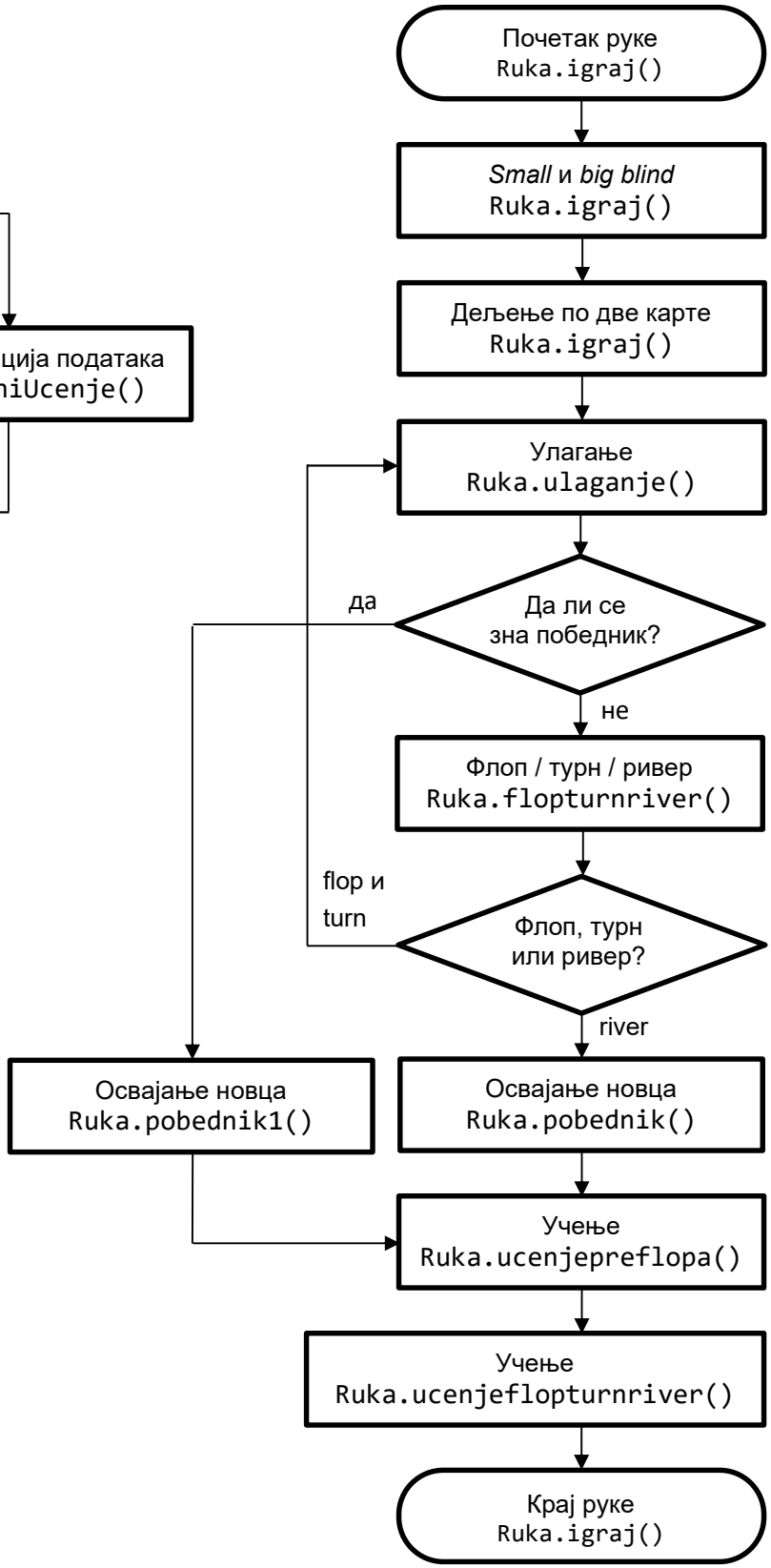
Програм се састоји из следећих класа и њихових важних функција:

1. Main
   - main
   - PocniUcenje
   - UcitajPodatke
   - IspisiPodatke
2. Cashgame
   - igraj
3. Ruka
   - igraj
   - ulaganje
   - flopturnriver
   - pobednik
4. Igrac
   - preflop
   - flopturnriver
   - preflopflopturnriver
   - ucenjepreflopa
   - ucenjeflopturnriver
5. Verovatnoca
6. Istorija
   - parametripreflop
   - parametriflopturnriver
7. Karta
   - dvekarte
8. Spil

На следећим дијаграмима су приказани рад програма и играње једне руке:



Дијаграм 2: Ток програма



Дијаграм 3:
Ток руке у програму

5.1. Припрема

Програм почиње `main` функцијом класе `Main`. Ако се програм покреће први пут, или је потребно почети машинско учење од почетка, покреће се функција `PocniUcenje`, у којој се иницијализују матрице вредности стања. Иначе, покреће се функција `UcitajPodatke`. Константа `pracenje` служи да одредимо како желимо да пратимо рад програма. Затим се покреће функција `igraj` у класи `Cashgame`. Параметром одређујемо колико руку желимо да програм одигра; може бити и бесконачно. У функцији се додају играчи и покреће се петља у којој се игра жељени број руку. Резултати се исписују у функцији `IspisiPodatke` класе `Main`. Променљива `ii` одређује колико често желимо да испишемо резултате. Пожељно је исписивати ретко зато што одузима време. После одигране руке у функцији `igraj` класе `Ruka`, проверава се да ли је играч испунио услове да напусти игру.

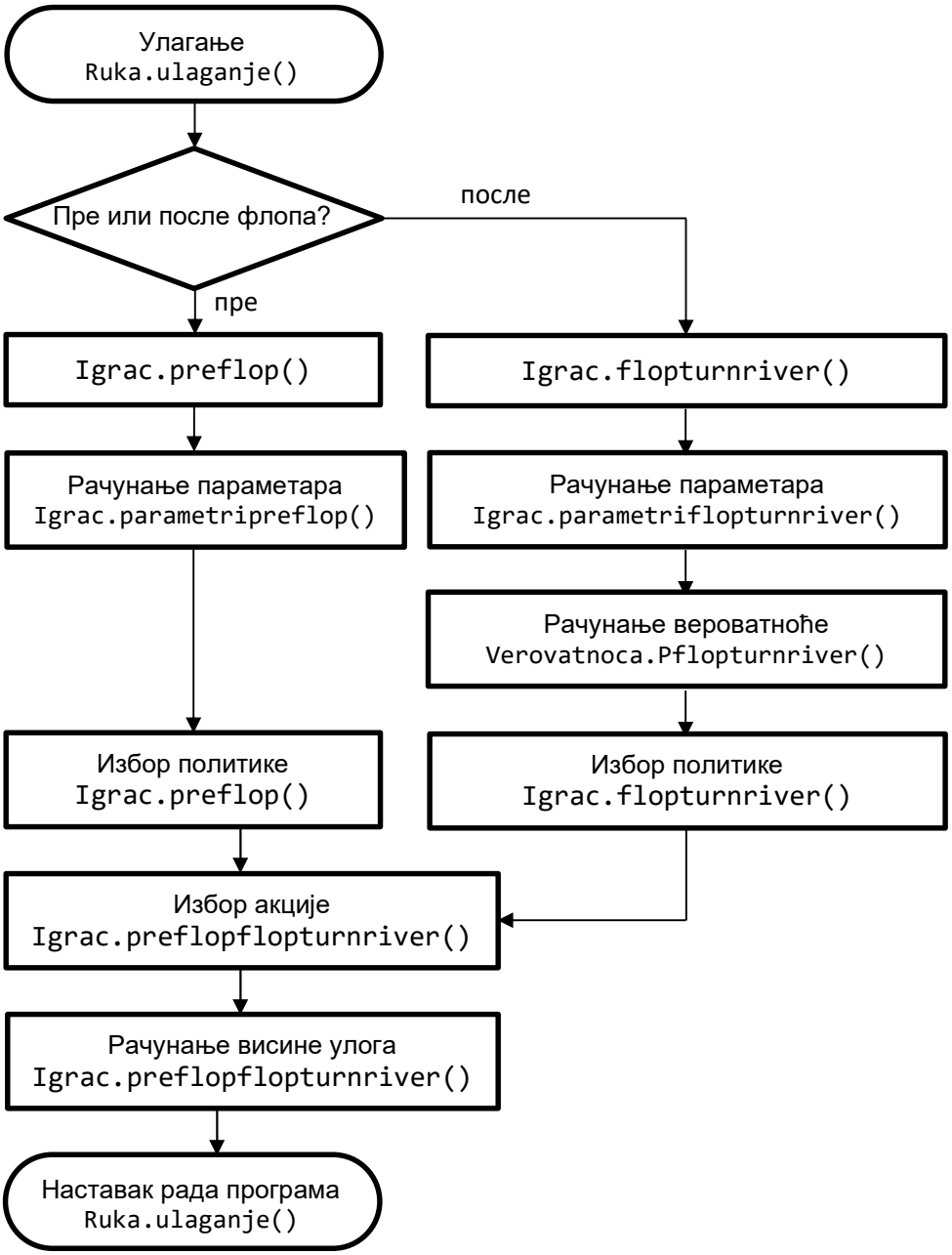
5.2. Рука

Функције `igraj` и `flopturnriver` класе `Ruka` у потпуности прате ток руке у игри. На стандардни излаз се исписује новчано стање свих играча и ко улаже *small* и *big blind* уз карте које има.

Улагање се врши у истоименој функцији. Акција, тј. величина улога се бира у функцијама `preflop` и `flopturnriver` класе `Igrac`. Она се затим обрађује у зависности од типа (*fold*, *check*, *call*, *bet* или *raise*) и на крају исписује на стандардни излаз.

Ако се рука завршила пре показивања карата, тј. сви играчи осим једног су се повукли, покреће се функција `pobednik1`, а иначе функција `pobednik`. У првом случају победник осваја новац, а у другом се руке сортирају према јачини и затим победници освајају новац. Јачине руку се одређују функцијом `ruka` класе `Verovatnoca`. На стандардни излаз се исписује ко је колико новца освојио и са којим картама.

5.3. Одређивање висине улога



Дијаграм 4:
Одређивање висине улога

5.3.1. Параметри

У зависности од фазе руке, улази се у функцију `preflop` или `flopturnriver` класе `Igrac`. Рачунају се параметри атрибута у функцијама `parametripreflop` и `parametriflopturnriver` класе `Istorija` у складу са табелом у поглављу „Упрошћавање модела“.

Не треба нам тачна вредност параметара вероватноћа, укупан број играча, број играча у игри и позиција, већ њихова ефективна вредност која зависи од односа новца играча. На пример: први играч има 2000$, други 1000$, а трећи 100$. Пре него што се поделе карте, вероватноћа да било ко добије је 1/3. Међутим, први играч не гледа трећег играча као равноправног себи зато што не може да узме озбиљну количину новца, већ се фокусира на другог. Такође, он може да освоји највише 1000$ од другог играча, па је ефективна вредност његовог новца 1000$. Ако трећи играч победи, првом и другом играчу је остало по 900$ које ће да освоји онај ко је други најбољи. Зато први играч рачуна следећи укупан број играча: 1000/1000 + 1000/1000 + 100/1000 = 2,1. Исто важи и за број играча у игри, позицију и вероватноћу. На пример, пре почетка руке вероватноћа је 1/2 против једног и 1/3 против два играча, па је ефективна вероватноћа (1/3*(100*2) + 1/2*(900*1)) / 1100 = 0,4697.

Овај алгоритам се извршава у функцијама `ocekivanje`, `ocekivanje1` и `pozicija` у класи `Ruka` и у функцији `Pflopturnriver` у класи `Verovatnoca`.

5.3.2. Вероватноћа

У функцији `flopturnriver` класе `Igrac` се користи вероватноћа израчуната у функцији `Pflopturnriver` класе `Verovatnoca`. Тачна вероватноћа се рачуна у функцијама `Pflop`, `Pturn` и `Priver` која се затим коригује по алгоритму из поглавља „Параметри“.

Идеја рачунања вероватноће је једноставна: преброје се све победничке, ремизирајуће и губитничке комбинације. Међутим, петља не пролази кроз све карте, зато што су боје карата еквивалентне, већ само кроз све различите случајеве броја карата исте боје уз множење бројем понављања за друге распореде боја.

У функцијама `Pflop`, `Pflop1`, `Pturn` се пролазе све могуће карте или комбинације карата које могу да буду извучене до краја руке, а затим се рачуна вероватноћа. У функцији `Priver` се само рачуна вероватноћа јер су све карте већ извучене.

У функцијама `Pflop1`, `Pturn1` и `Priver` се, за дату руку од 7 карата, позива функција `Protivnikovekarte`, у којој се пролазе све комбинације две карте противника. Коначно, у функцији `Protivnikovekarte1` се упоређују могућа или коначна рука играча са могућом руком противника у функцији `rukaboja` или `rukabezboje`.

5.3.3. Политика и избор акције

Сви израчунати параметри се уписују у променљиву `istorija`. На основу њих се бира одговарајућа политика – низ из матрице вредности стања. Елементи низа су различите акције одређене у поглављу „Акције“.

Како би се учење убрзало, вредности сваке акције се додаје по половина вредности претходне и следеће акције. Додатно, после флопа се не узима у обзир само тачна политика,

већ и суседне политике за претходни и следећи интервал вероватноће. Тачној политици се додају половине вредности суседних политика.

Тиме се постиже следеће: Суседне политике и награде за суседне акције морају бити сличне због великог броја акција и стања атрибута вероватноћа, па се коришћењем већег броја података постиже брже учење. Са друге стране, ако је вредност одређене акције добијена претежно на основу руке која је завршена „на срећу“, тј. дошла је баш она карта која је била потребна, умањује се значај тога.

На крају, вредност акције h се добија на следећи начин:

```
politika[h]            + politika[h-1]/2          + politika[h+1]/2 +
prethodnapolitika[h]/2 + prethodnapolitika[h-1]/4 + prethodnapolitika[h+1]/4 +
sledecapolitika[h]/2   + sledecapolitika[h-1]/4   + sledecapolitika[h+1]/4
```

Затим се у функцији `preflopflopturnriver` бира акција на основу модификоване политике. Акција се разматра само ако је могућа по правилима: једнака или бар дупло већа од улога који је потребно испратити. Ако се дође до акције која никад до тада није била изабрана, она се бира без обзира да ли постоји боља. Иначе, саставља се скуп акција које су до на 0,1 добре као и најбоља акција. Затим се са вероватноћом 0.05, 0.1, 0.2 или 0.3, у зависности од фазе руке, била случајна, иначе се бира најбоља акција. Акција као индекс Фибоначијевог низа се чува у променљиви `akcije`. Онда се на основу Фибоначијевог низа рачуна висина улога. На крају се проверава да ли задовољава правила игре, по потреби се коригује, и повратна вредност функције је укупан улог играча у тренутној фази руке, тј. рачунају се и сви претходни улози.

5.4. Учење

Учење почиње иницијализацијом матрица вредности стања у функцији `PocniUcenje` класе `Main`. Кад се играч повуче из руке или кад се рука заврши, позивају се функције `ucenjepreflopa` и `ucenjeflopturnriver`. У њима се прво рачуна награда. Затим се, на основу параметара сачуваних у променљиви `istorija`, бира исти низ из матрице вредности (политика) који је био основа за кориговану политику на основу које се бирала акција. На крају се ажурира вредност политике на позицији записаној у променљиви `akcije` по алгоритму Sarsa.

Тиме се завршава рука и програм се враћа у функцију `igraj` класе `Cashgame`.

6. Анализа резултата

Због великог скупа стања потребно је много времена да програм научи да игра квалитетно. Како нам толико време није на располагању, не можемо проверити да ли је играч довољно добар да може да игра успешно против осредњих или добрих покераша. Из истог разлога не можемо да оцењујемо појединачне акције на основу искуства играња покера.

6.1. Упоређивање квалитета игре

Једино што можемо је да проверимо да ли програм учи да игра, тј. да ли временом постаје бољи. То се може урадити упоређивањем играча који користе податке добијене после различитог времена учења. Изабрани су три дана и два месеца[9]. Подаци добијени после три дана су оскудни због великог скупа стања и пуно акција се бира на случајан начин, док се, после два месеца учења, много мање акција бира на случајан начин. Није могуће упоредити два играча тако да се код оба велика већина акција не бира на случајан начин због недостатка времена.

Како играју 10 играча, 5 играча користе податке добијене после 3 дана учења. Очекујемо да играчи који користе тачније податке зараде више новца.

И заиста, после једног дана рада програма, играчи од три дана су сакупили 113,290,000 $, а играчи од два месеца 353,000,500 $, што значи да је програм остварио напредак за два месеца учења.

[9] Број играча мора бити 10. Да би се спровело успешно тестирање неопходно је поделити играче на тимове са истим бројем играча. Како је тим од два играча мали, јер неке ситуације са малом вероватноћом могу пореметити једног играча на дуже време и тиме утицати на коначан исход, остају два тима од 5 играча.

7. Биографија

Филип Пешић је рођен 21.3.1994. у Београду.

Завршио је основну школу Павле Савић 2009. године, Математичку гимназију 2013. године и Математички факултет 2017. године.

Бави се програмирањем.

8. Литература

[1] Richard Sutton, Andrew Barto: *Reinforcement Learning: An Introduction*, 2017

[2] Комплетан доказ се може наћи овде:
https://www.cse.iitb.ac.in/~shivaram/resources/ijcai-2017-tutorial-policyiteration/tapi.pdf

[3] На пример, на овом сајту се налази on-line покер школа:
https://www.pokerstrategy.com/strategy/