UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Marko Drevenšek

Kombinatorična igra filozofov nogomet

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

MENTOR: izr. prof. dr. Gašper Fijavž

Ljubljana, 2016

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Filozofov nogomet (angl. Philosopher's football, tudi Phutball) je kombinatorična igra za dva igralca na igralni plošči. V delu obravnavajte njene lastnosti s stališča računske zahtevnosti in izdelajte spletno aplikacijo za njeno igranje na igralnih ploščah različnih dimenzij.

IZJAVA O AVTORSTVU diplomskega dela

Spodaj podpisani Marko Drevenšek, z vpisno številko 63050027, sem avtor diplomskega dela z naslovom:

Kombinatorična igra filozofov nogomet

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom izr. prof. dr. Gašperja Fijavža
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 8. septembra 2016 Podpis avtorja:

Zahvaljujem se mentorju izr. prof. dr. Gašperju Fijavžu za vodenje, nasvete in vso pomoč pri izdelavi diplomske naloge.

Zahvaljujem se tudi družini in prijateljem, ki so mi stali ob strani.

Kazalo

Seznam uporabljenih kratic

kratica	Angleško
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
CNF	Conjunctive Normal Form
3-SAT	3-satisfiability problem

Povzetek

Naslov: Kombinatorična igra filozofov nogomet

Kombinatorične igre so igre, kjer igralca izmenično izvajata poteze. Pri igri nimamo nikakršnih pripomočkov, ki bi na igro vplivali naključno. Igralca imata popolno informacijo o preteklih potezah za odločanje, kako igrati naprej. Pravila so takšna, da je igra končna.

V delu predstavimo malo znano igro *filozofov nogomet* oziroma *figomet*. Predstavimo, zakaj je igra težka, in zakaj je odločiti, ali lahko igralec v dani situaciji zmaga z eno potezo, NP-poln problem. Dokazovanja smo se lotili s prevedbo že znanega NP-polnega problema 3-SAT.

Poleg tega smo implementirali igro kot aplikacijo za dva igralca, ki je dostopna preko spletnega brskalnika.

Ključne besede: kombinatorična igra, računska zahtevnost, filozofov nogomet, NP-polnost, 3-izpolnljivost, spletna aplikacija.

Abstract

Title: Combinatorial game Philosopher's Football

Combinatorial games are games where the players alternately take moves. In the game we do not have any chance devices that could impact on the game randomly. Players have complete information about past moves to decide how to play on. The rules are such that the game must eventually end.

Philosopher’s football or Phutball is a not so well-known combinatorial game which is the focus of this work. We present why the game is difficult and why determing whether a player can win in a given situation in one move is an NP-complete problem. We prove it by reducing to an already known NP-complete problem 3-SAT.

We have also implemented Phutball application for two players, which is accessible through a web browser.

Keywords: combinatorial game, computational complexity, Philosopher's football, NP-completeness, 3-SAT, web application.

Poglavje 1 Uvod

Teorija kombinatoričnih iger je veja uporabne matematike in teoretičnega računalništva, ki se običajno ukvarja z igrami, kjer igralca izmenično izvajata poteze ter imata popolno informacijo o preteklih potezah, ki jima pomaga pri odločanju, kako igrati naprej.

Poznamo veliko različnih kombinatoričnih iger za dva igralca, med katerimi smo izbrali nekaj zanimivih in opisali njihove značilnosti, ki so pomembne za računsko zahtevnost. Najbolj znana med njimi je igra *šah*. Igra *filozofov nogomet* ni tako poznana in razširjena, zato smo se odločili, da poizvemo čim več znanega o njej. Izbrali smo krajše ime *figomet*.

V naslednjem poglavju bomo omejili *figomet* na enorazsežno igralno ploščo. Na primerih bomo pokazali kakšne so strategije za zmago in dokazali njihovo pravilnost.

Nato bomo predstavili, zakaj je igra *figomet* težka in zakaj je odločiti ali lahko igralec v dani situaciji zmaga z eno potezo NP polni problem. Dokazovanja se bomo lotili s polinomsko prevedbo že znanega NP-polnega problema 3-SAT.

Na koncu bomo pregledali tehnologije potrebne za implementacijo *figometa* kot aplikacijo za dva igralca. Prav tako bomo predstavili uporabniški vmesnik.

Poglavje 2 *Figomet* in kombinatorične igre

Pri analizi kombinatoričnih iger se pojavljata dve glavni vprašanji:

- Ali lahko v dani situaciji zmagamo?
- Katere so poteze, s katerimi lahko zmagamo?

2.1 Teorija kombinatoričnih iger

Tako kot večina ostalih iger imajo kombinatorične igre igralce, poteze, pravila, zmagovalce in poražence. Obstajajo pa tudi igre, ki nimajo vedno zmagovalca. Šah, na primer, se lahko konča z remijem. Osredotočili se bomo na igre za dva igralca, čeprav imajo lahko kombinatorične igre več kot dva igralca ali samo enega. Prav tako nas zanimajo igre, ki se vedno zaključijo.

Teorija kombinatoričnih iger je veja uporabne matematike in teoretičnega računalništva, ki po navadi raziskuje igre, kjer igralci vlečejo poteze izmenično, hkrati pa vsi igralci v dani situaciji poznajo vse pretekle in možne poteze. V veliki meri se študije omejujejo na dva igralca, kjer imamo vedno zmagovalca. Ker matematične tehnike napredujejo, se s tem veča tudi polje raziskav. Obravnavamo lahko tudi igre na srečo, igre z nepopolno informacijo ter igre, kjer lahko igralci izvajajo več potez hkrati. Med kombinatorične igre štejemo tudi igre za enega igralca, kot so *sudoku, minolovec, pasjansa*,…

Za *nepristranske kombinatorične igre* velja, da so dovoljene poteze odvisne le od trenutne situacije v igri in ne od tega, kateri igralec je na vrsti. Ta opredelitev je bila v veliki povezavi ob nastajanju teorije o kombinatoričnih igrah. Ena izmed takih zelo pomembnih iger je *nim*. To je igra za dva igralca, kjer velja pravilo, da igralec izgubi, ko več ne more izvesti poteze. V tridesetih letih sta Sprague [14] in Grundy [8] neodvisno pokazala izrek, da lahko vse nepristranske igre predstavimo s kupi v igri *nim* in tako posplošimo in poenotimo kombinatorične lastnosti strategij teh iger. Kasneje v šestdesetih so Berlekamp, Conway in Guy obravnavali igre, pri katerih dopustne poteze enega igralca niso vedno dopustne tudi za drugega. Rezultati so predstavljeni v njihovi knjigi Winning Ways for your Mathematical Plays leta 1982 [2].

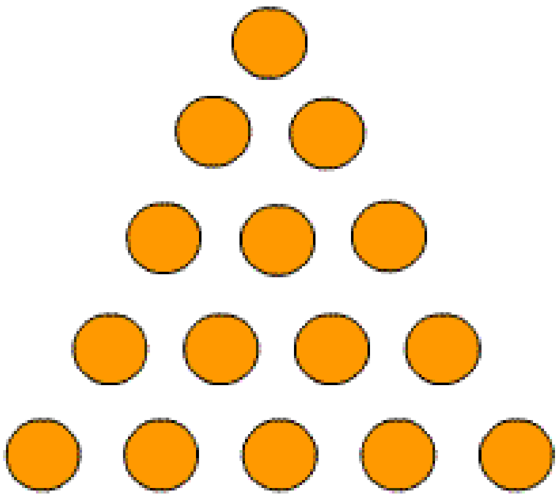
Kombinatorične igre so običajno predstavljene v obliki, kjer igralec zmaga, ko njegov nasprotnik nima več razpoložljivih potez. Vsako kombinatorično igro lahko pretvorimo v to obliko, če le ima naša igra zgolj dva različna možna konca. Eden izmed najbolj pomembnih konceptov teorije kombinatoričnih iger je, da je vsota dveh iger nova igra, v kateri lahko vsak igralec izvede potezo v prvi ali drugi igri in v vsaki situaciji. Igralec pa zmaga, ko nasprotnik nima več razpoložljivih potez v nobeni od obeh iger. Takšno kombiniranje iger vodi do bogate in močne matematične teorije. Teorija pristranskih iger Johna Conwaya [2] je bila inspiracija pri opazovanju igre *go*, ki je lahko po navadi razbita na vsoto več enostavnejših končnih iger, ki so izolirane med seboj na različnih delih igralne plošče.

2.2 Igre za dva igralca

Pri vsaki igri določimo nabor možnih potez dveh igralcev, ki ju imenujemo Levi in Desni. Novo pozicijo igre, ki jo dobimo z izvedbo poteze, lahko obravnavamo kot novo igro. Ta ideja, da vidimo igre kot možne poteze drugih iger vodi do rekurzivne matematične definicije iger, kar je standardni pristop v teoriji kombinatoričnih iger. V tej definiciji ima vsaka igra notacijo {L | D}, kjer je L množica pozicij v igri, ki jih lahko izvede Levi igralec in D množica za Desnega igralca.

2.2.1 Nim

Je matematična igra strategije, kjer dva igralca izmenično odstranjujeta predmete iz kupov. Na vsaki potezi mora igralec odstraniti vsaj en predmet, lahko pa tudi več, denimo največ pet, predmetov iz istega kupa. Za oba igralca so vedno možne iste poteze, odvisno je le, kdo je na vrsti. Zmaga tisti igralec, ki odstrani zadnji predmet, zato oba stremita k istemu cilju. Vedno zmaga en ali drugi igralec, saj število predmetov v kupu ali kupih strogo pada.

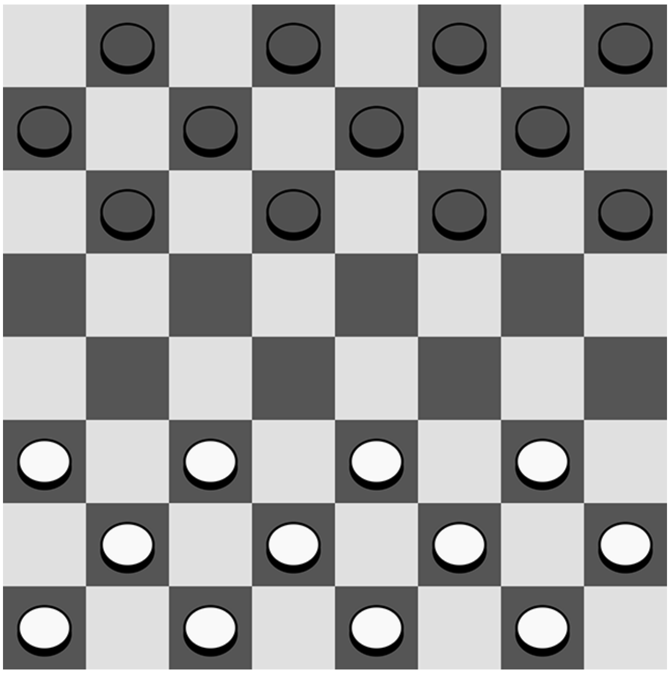


Slika 2.1: Primer igre *nim* z enim kupom.

Za igro *nim* z več kupi in predmeti velja, da lahko v vsaki dani situaciji napovemo, ali ima igralec na potezi zmagovalno strategijo. Poleg tega lahko napovemo tudi, katere poteze so zmagovalne.

2.2.2 Dama

Dama je družabna igra, ki se igra na igralni plošči velikosti 8x8. Igralca imata vsak svoje kamne in dovoljene diagonalne skoke.



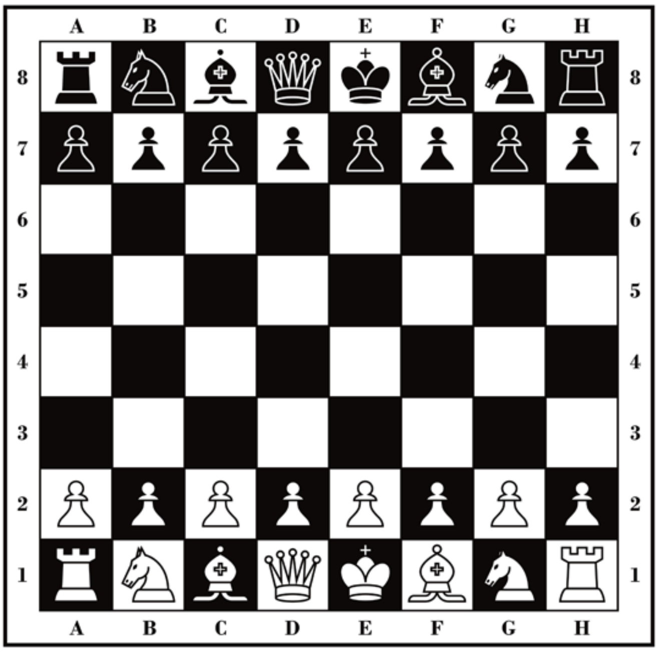
Slika 2.2: Postavitev kamnov začetne pozicije pri igri *dama* [15].

V dani situaciji imata oba različne možne poteze, saj lahko premikata samo svoje kamne. Prav tako imata vsak svoj cilj, saj poskušata odstraniti vse nasprotnikove kamne.

2.2.3 Šah

Šah je igra na igralni plošči 8x8, kjer ima vsak igralec 16 figur. Obstaja 6 različnih tipov figur, ki se premikajo na različne načine. Vsak igralec lahko premika samo svoje figure. Cilj igre je matirati nasprotnika. To dosežemo tako, da kralja, nasprotnikovo figuro, spravimo v položaj, kjer se ne more izogniti zajetju. Igra se lahko na več načinov konča z remijem, neodločenim izidom.

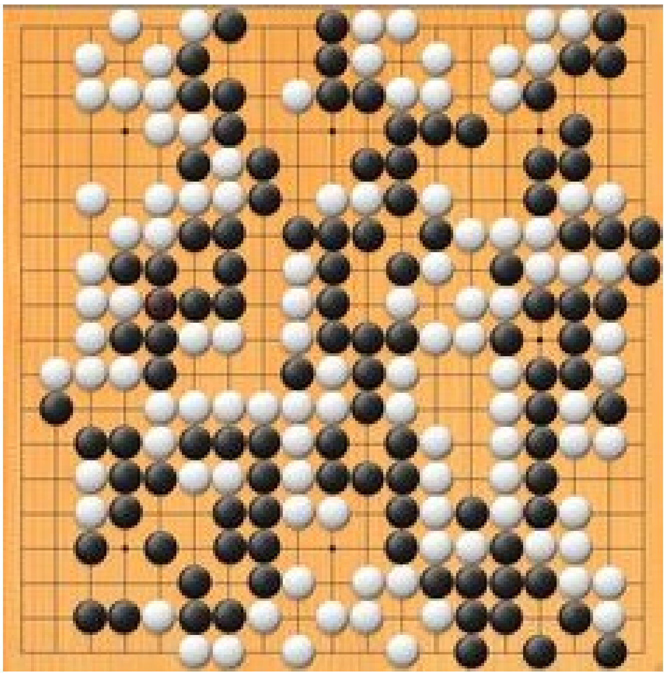
Število legalnih potez pri šahu je približno 10^{43} [3]. V tipični situaciji imamo v povprečju od 30 do 40 možnih potez, vendar je to število lahko tudi 0 (v primeru mata) ali kar 218 [5]. V šahovski terminologiji ena poteza ustreza dvema kombinatoričnima potezama, po eni belega in črnega igralca. *Razširjeni šah*, za katerega velja, da igramo na večji igralni plošči velikosti *n* x *n* z enim kraljem za vsakega igralca, ter večje število ostalih pet tipov figur, je EXPTIME-poln [5].



Slika 2.3: Igralna plošča s figurami pri igri *šah* .

2.2.4 Go

Igro *go* igramo na igralni plošči velikosti 19x19, kjer ima en igralec bele kamne in drugi črne kamne. Vsak igralec lahko polaga kamne svoje barve na ploščo in ne more odstraniti nobenega svojega kamna. Če igralec s svojimi kamni obkroži iz vseh smeri skupino nasprotnikovih kamnov, jih (nasprotnikove kamne) odstrani iz igralne plošče. Igralca se izmenjujeta v potezah, igra pa se zaključi, ko nihče od igralcev ne želi več izvajati potez. Zmaga tisti, ki ima na koncu več kamnov. Z računskega stališča je *go* veliko bolj kompleksna igra in ima več kombinacij kot šah, čeprav so pravila relativno enostavna.



Slika 2.4: Igralna plošča 19x19 pri igri *go* 0.

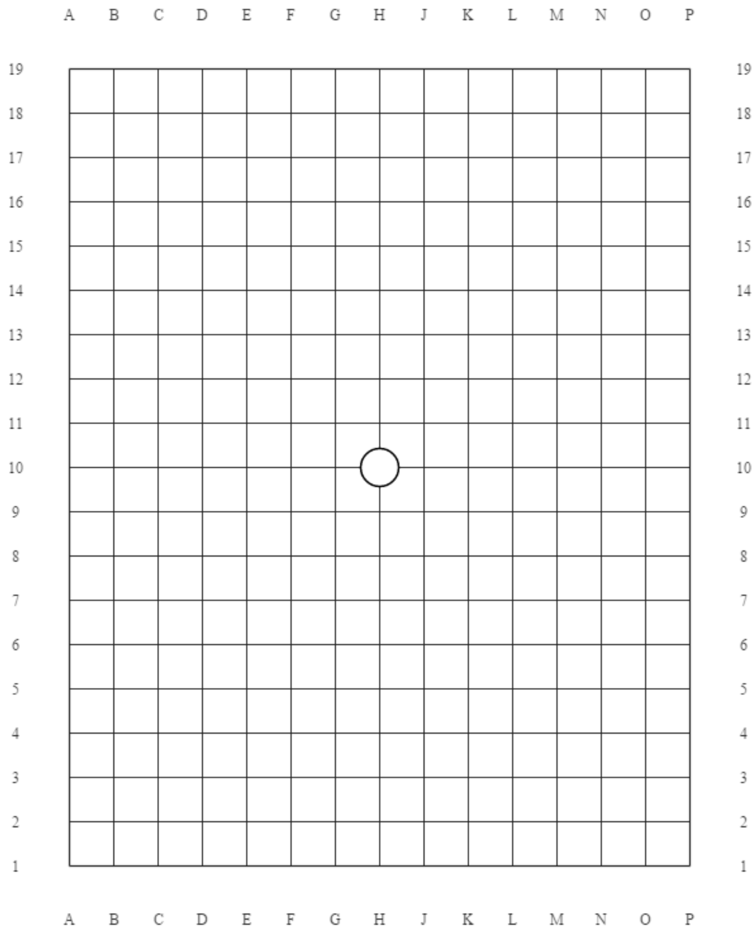
2.3 Figomet

Filozofov nogomet (angl. Phutball ali tudi Philosopher's Football) je kombinatorična igra za dva igralca, ki so jo opisali Elwyn Berlekamp, John Horton Conway in Richard K. Guy v knjigi *Winning Ways for your Mathematical Plays* [1]. Skušali smo najti primerno ime za to zanimivo igro. Nogomet je zelo priljubljen šport, zato smo ohranili ta del v imenu in ga skrajšali na *figomet*, kar bomo uporabljali tudi v preostanku besedila.

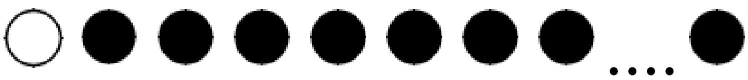
2.3.1 Pravila

Za igranje potrebujemo:

- Igralno ploščo velikosti 15x19 (oziroma *m* x *n* v posplošeni varianti):



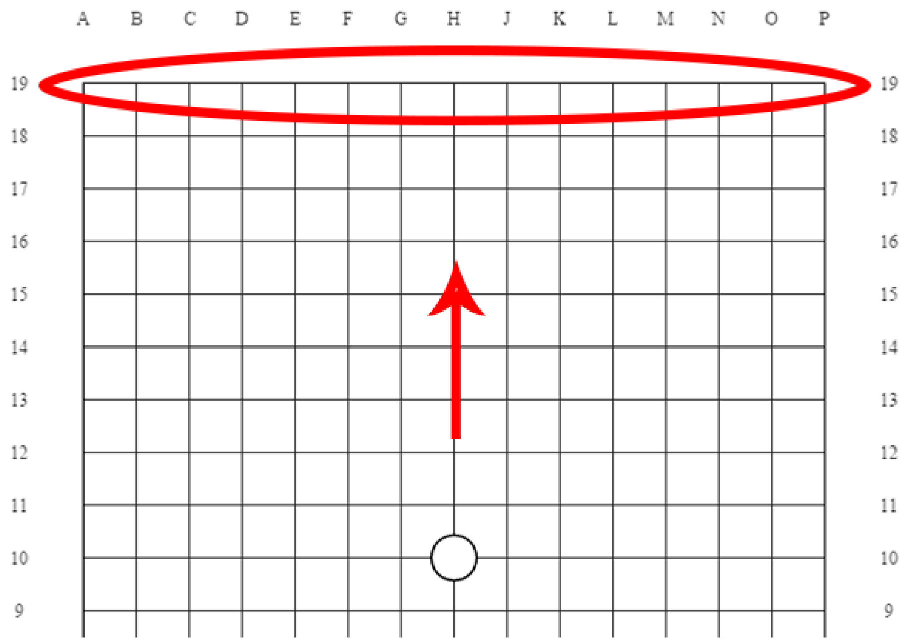
- Beli kamen in več črnih kamnov:



Igramo na plošči velikosti 15x19. Stolpce na plošči označimo s črkami od A do P (standardno preskočimo črko I) z leve proti desni s in številkami od 1 do 19 od spodaj navzgor. Igramo s črnimi in belim kamnom. Beli kamen imenujemo *žoga* in je isti za oba igralca. Črni kamni predstavljajo igralce, ki jih dodajamo na igralno ploščo in jih bomo imenovali kar *kamni*. Po navadi igramo kar s pomočjo igralne plošče za igro *go*, ki je velikosti 19x19, saj je zelo težko dobiti ploščo nestandardnih velikosti, predvsem nesimetrično. Za igro uporabimo tudi beli kamen in črne kamne, ki so v kompletu igre *go*.

Na začetku igre imamo postavljeno žogo na sredini plošče, na poziciji H10. Za prvega igralca je gol zgornja vrstica od A19 do P19, za drugega pa vrstica od A1 do P19. Žogo poskušamo s skoki spraviti na vrstico gola ali s skokom čez njo, da dosežemo gol. Tako prvi igralec poskuša spraviti žogo navzgor, kot je je ilustrirano spodaj.

Gol za prvega igralca:



Igralec ne sme spraviti žoge izven igralne plošče na levi in desni strani. Kamne se dodaja na križišče črt in ne na polja znotraj teh črt, kot velja za šah. Igralca sedita nasproti drug drugega, na daljši stranici igralne plošče. Igranje izven plošče ni dovoljeno. Časovne omejitve pri igranju ni in načeloma lahko pozicije v igri ponavljamo, tako da lahko igra traja zelo dolgo, tudi več ur.

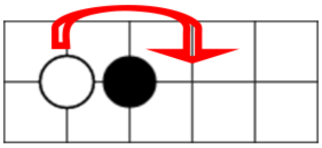
Igralec, ki je na vrsti ima na voljo dve različni potezi:

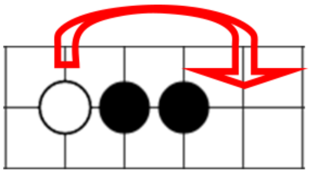
- Dodajanje kamna,
- skok z žogo.

Igralca poteze opravljata izmenično. Igralec ki je na vrsti, mora bodisi dodati kamen bodisi skočiti z žogo. Potezo mora tudi odigrati, saj preskok poteze ni dovoljen. Dovoljeni potezi sta dodajanje kamna na nezasedeno mesto (križišče črt) oziroma skok žoge preko kamna, ali več zaporednih kamnov. Skok in dodajanje kamnov ni dovoljen izven plošče. Izjema je le zadnji skok preko kamna na črti gola, saj je dodajanje kamnov tudi na to mesto dovoljeno.

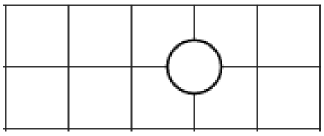
Kamen lahko dodamo na nezasedeno mesto, ter si tako omogočimo potencialni skok žoge. Kamnov, ki so dodani na plošči med seboj ne razlikujemo, saj so enaki za oba igralca. Tako je pri dodajanju potrebno paziti, da s tem, ko dodamo kamen, nasprotnemu igralcu ne omogočimo premoči v igri. Vsak igralec ima na voljo eno potezo, ki predstavlja postavljanje kamna. Ker v eni potezi ni dovoljeno dodajanje več kamnov, je po izvedbi na vrsti drugi igralec.

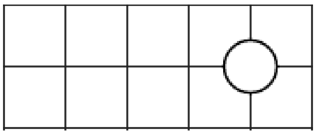
Skok z žogo lahko izvedemo, ko imamo na sosednjem mestu kamen. Skok lahko izvedemo tudi preko več zaporednih kamnov.



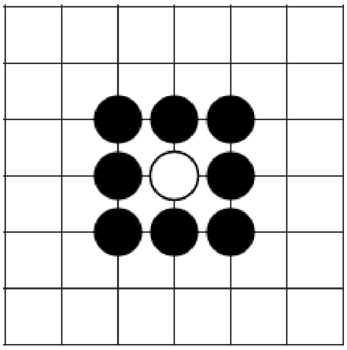


Po skoku se kamen oz. več kamnov, ki smo jih preskočili, odstrani.

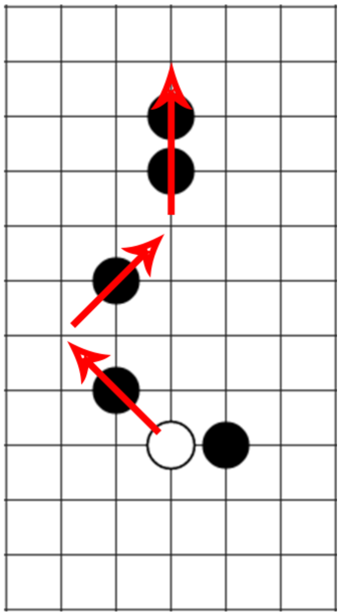




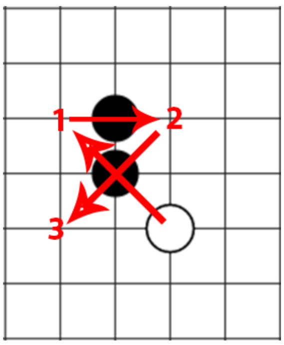
Skačemo lahko v vseh osem smeri, vodoravno, navpično in diagonalno. Prikazujemo situacijo, ker lahko z žogo skočimo v vsako od osmih smeri (in preskočimo po en kamen):



Po vsakem skoku lahko **nadaljujemo** s skokom, če za žogo na novem mestu velja pogoj za skok, ki smo ga opisali na začetku. Nadaljnji skok ni obvezen, saj lahko prepustimo potezo nasprotniku. Skoki niso mogoči izven igralne plošče, tudi če je skok samo del zaporedja skokov.



Opozoriti je potrebno, da se kamni odstranijo takoj po skoku. Opisani skok v spodnjem primeru ni mogoč.



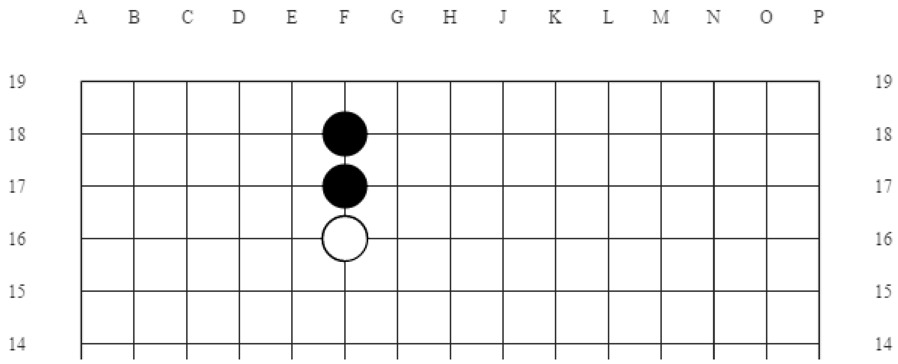
Najprej skočimo na mesto označeno s številko 1, kamen se odstrani takoj. Nato skočimo na mesto 2, ponovno se odstrani preskočeni kamen. Ko smo z žogo na mestu 2 na igralni plošči ni več nobenega kamna, tako da nadaljnji skok ni več mogoč. Ta omejitev se pojavi, saj kamne vedno odstranimo takoj po skoku, tudi če skok nadaljujemo.

2.3.2 Cilj

Cilj igre:

- Zadeti gol.

Igra se začne, ko postavimo žogo na sredino igralne plošče. Igralec lahko izvaja dve različni potezi, s katerimi poskuša zadeti gol. Za prvega igralca je gol zgoraj in za drugega spodaj. Igralca se skozi igro poskušata z žogo približati golu in ga tudi zadeti. Igralca imata različen cilj, se pravi zadeti vsak nasprotnikov gol. Primer zmage prvega igralca, če izvede skok:

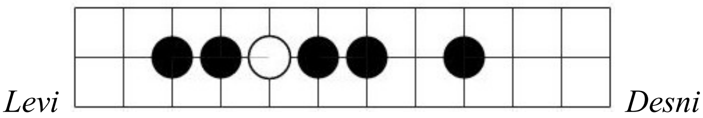


Teoretično je možno, da se igra povrne v prejšnjo pozicijo, zato bi bilo načeloma potrebno, da dodamo pravilo za izogibanje zanke, kot ga poznamo v *šahu* in igri *go*. Vendar se po eni strani ponovljena pozicija v praksi ne pojavlja ravno pogosto, po drugi strani pa človeška igralca ponovitev pozicije tudi zelo težko zaznata.

Poglavje 3 *Enorazsežni figomet*

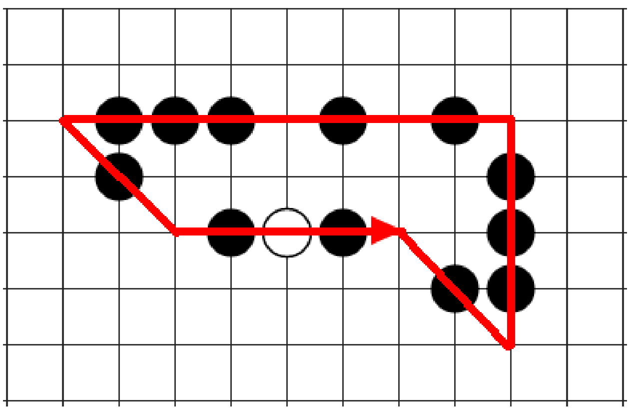
Enorazsežni figomet igramo na traku končne dolžine. Pravila za dodajanje kamna in skoka z žogo ostajajo enaka, vendar lahko skačemo samo levo in desno. Cilj igre se primerno spremeni, tako Levi igralec zmaga, če spravi žogo do ali preko skrajno desnega praznega prostora, ki ga imenujemo desni gol. Obratno velja za Desnega igralca. Na kratko bomo Levega igralca imenovali *Levi* in Desnega igralca *Desni*.

Spodaj je primer pozicije, kjer sta igralca že odigrala nekaj potez.

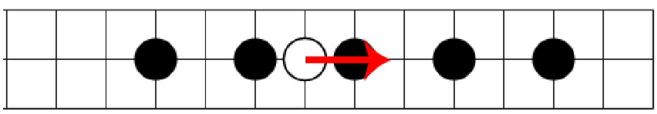


3.1 Analiza strategije za zmago

Omejitve pri igri se pojavijo predvsem pri skoku z žogo. Če pri *figometu* lahko skačemo v vse smeri, imamo pri enorazsežnem možnost skoka samo levo ali desno. Naj pokažemo, da je omejitev veliko večja. Vzemimo situacijo *figometa*, kjer lahko začnemo skok na desno in na koncu končamo na istem mestu kot smo začeli. Pri smeri skoka bomo uporabili notacijo smeri neba. Skoki žoge si sledijo vzhod, jugovzhod, sever, zahod, zahod, zahod, jugovzhod in vzhod. Pokazali smo, da kljub temu da smo skok začeli v smeri desno, smo se v nadaljnjih skoki lahko pomikali navzgor in nazaj levo. Spodaj je opisan primer.



Sedaj vzemimo podoben primer, vendar na enorazsežnem *figometu*.



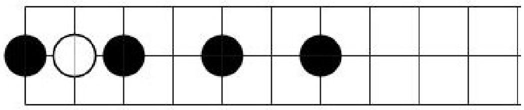
Skok lahko začnemo levo ali desno. Vendar lahko skok nadaljujemo samo v isti smeri, kot smo začeli. Nimamo možnosti, da se vrnemo nazaj, saj se kamen ki smo ga preskočili odstrani in skok lahko nadaljujemo samo naprej. Zdi se, da je zelo dobra strategija, da stremimo k temu, da se z žogo pomikamo proti golu. To lahko storimo tako, da dodajamo kamne in vedno skočimo najdlje kot je možno. Vendar to ni vedno res.

Tako se lahko v igri enorazsežnega *figometa* zgodi, da:

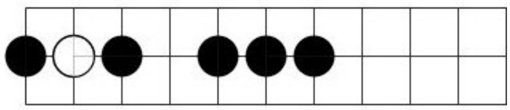
- Dodajanje kamnov na desno stran od žoge poslabša možnost zmage *Levega* in/ali
- odstranjevanje prostih mest desno od žoge poslabša možnost zmage *Levega*.

Prav taki trditvi veljata za *Desnega*, če smiselno prilagodimo smer igre. V spodnjih treh primerih bomo analizirali te trditve. V vseh primerih je na vrsti Levi in mora skočiti z žogo, saj drugače zmaga *Desni*.

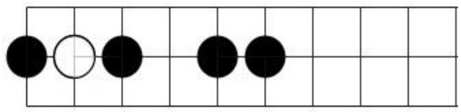
(1) Osnovna pozicija



(2) Dodali smo kamen na desno stran



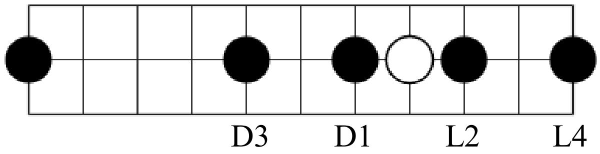
(3) Odstranili smo prazen prostor na desni

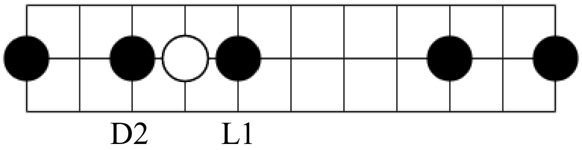


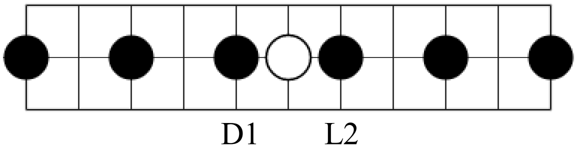
V vseh pozicijah (1), (2) in (3) mora *Levi* premakniti žogo, če ne želi izgubiti. V nasprotnem primeru namreč *Desni* z enim poskokom zabije gol. V poziciji (1) ima *Levi* tri možne poteze skoka.

Poteze skoka žoge in dodajanja kamnov prikažemo skupaj v eni sliki. Slike si ena pod drugo sledijo v zaporedju skokov. Skok žoge lahko razberemo iz nove pozicije žoge na sliki. Dodani kamni pa imajo spodaj označeno, kateri igralec jih je dodal in v kateri potezi so bili dodani. Če je kamen dodal *Levi* dodamo črko L oziroma črko D, če je kamen dodal *Desni*. Naslednja številka pa pomeni, katera poteza po vrsti je bila. Če želimo povedati da je *Desni* dodal kamen v tretji potezi bi zapisali: D3.

Levi skoči preko vseh treh kamnov (zmaga *Desni*):

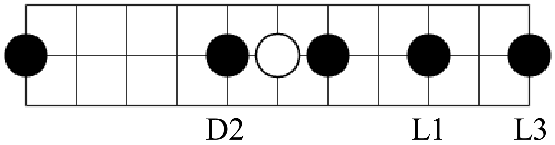


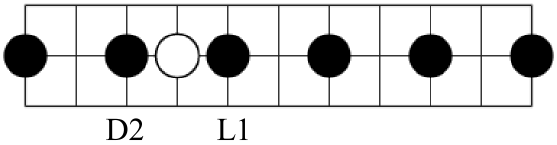




In v naslednji potezi *Desni* doseže gol.

Levi skoči preko dveh kamnov (zmaga *Levi*):

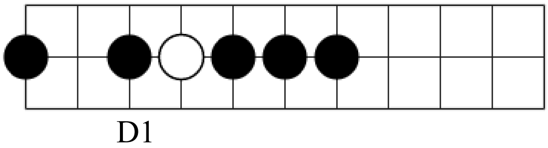


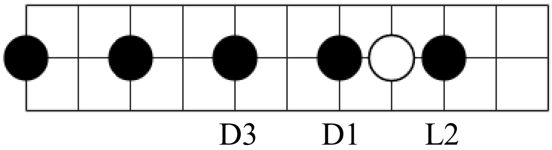


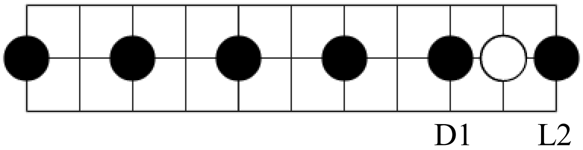
In v naslednji potezi *Levi* doseže gol.

Za primer (2) ne potrebujemo analize za skok preko vseh štirih kamnov, saj dobimo enako situacijo kot pri primeru (1), če skočimo preko vseh treh kamnov. Prav tako zmaga *Desni*.

Zato obravnavamo samo primer, ko *Levi* v prvi potezi skoči preko enega samega kamna (zmaga *Desni*):

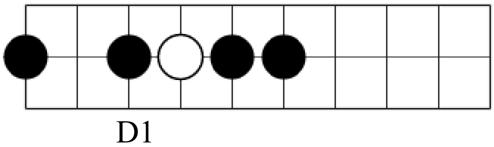


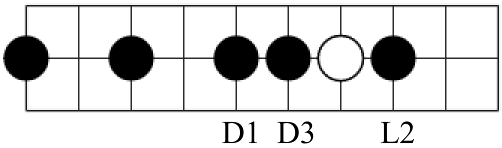


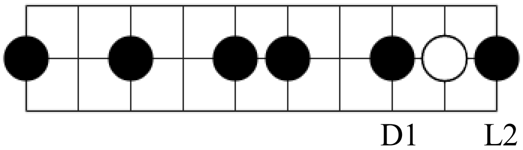


In v naslednji potezi *Desni* doseže gol.

Tudi za primer (3) ne potrebujemo analize za skok preko vseh kamnov, saj dobimo enako situacijo kot pri primeru (1), če skočimo preko vseh kamnov. Prav tako zmaga *Desni*. Zato pokažemo samo za skok preko enega kamna (zmaga *Desni*):

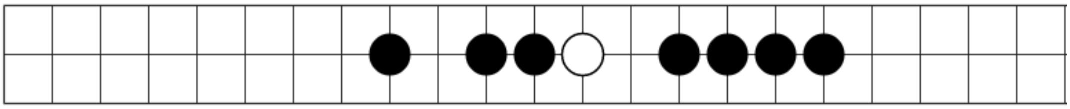




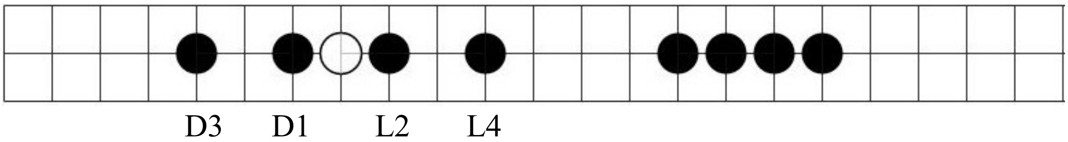


In v naslednji potezi *Desni* doseže gol.

V igri enorazsežnega *figometa* se nam zdi, da je edina zmagovalna strategija za *Levega* skok z žogo v desno. Vendar ta očitna strategija, da nikoli na skočimo nazaj in vedno skačemo samo proti golu, ne velja vedno. V spodnjem primeru je edina poteza, ki pripelje *Levega* do zmage, skok nazaj.

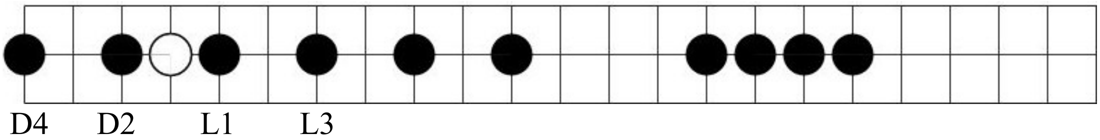


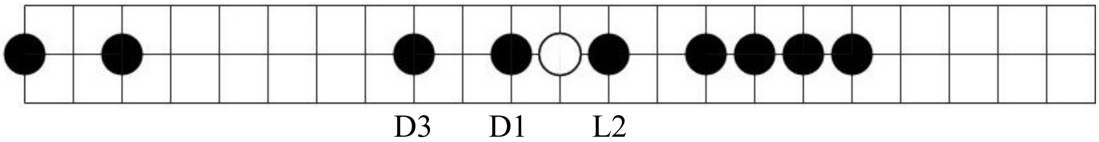
Zato skočimo nazaj:

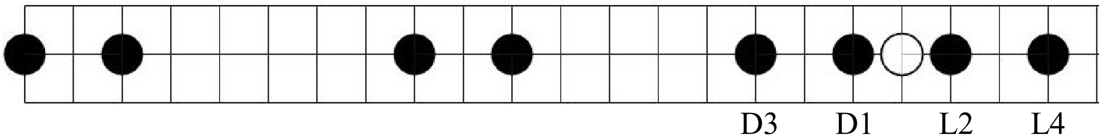


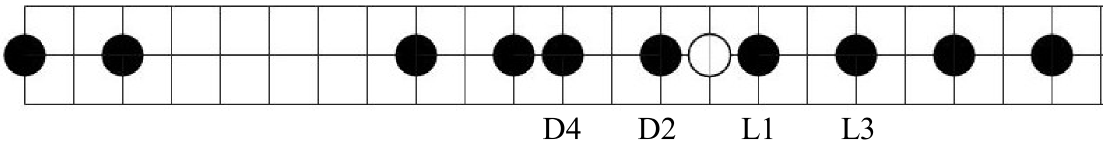
Nato imamo 2 možnosti:

Možnost 1 (*Desni* nadaljuje proti golu)



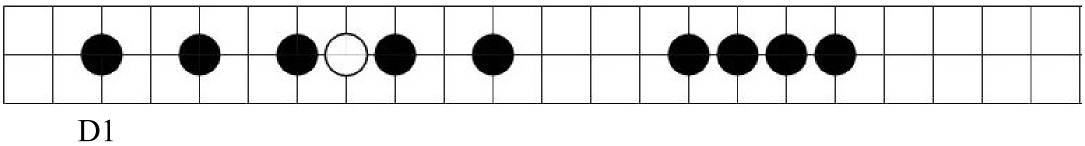


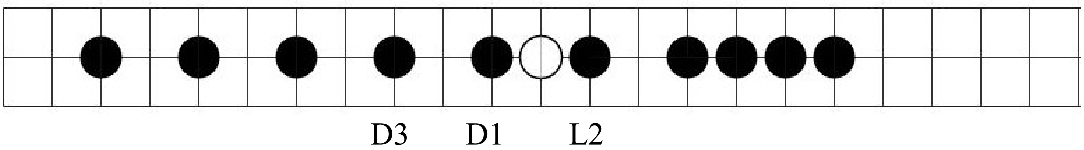


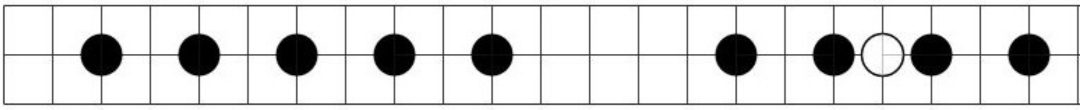


In v naslednji potezi *Levi* doseže gol.

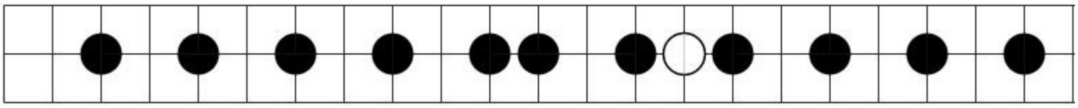
Možnost 2 (*Desni* doda kamen)







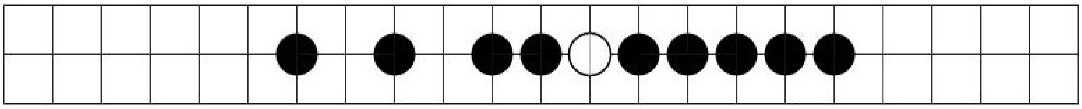
D3 D1 L2 L4



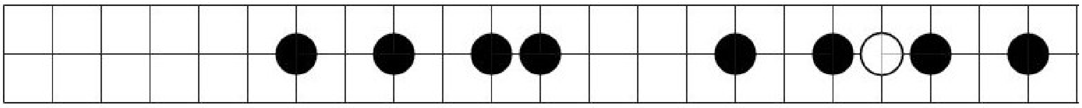
L3 L1 D2 D4

In v naslednji potezi *Levi* doseže gol.

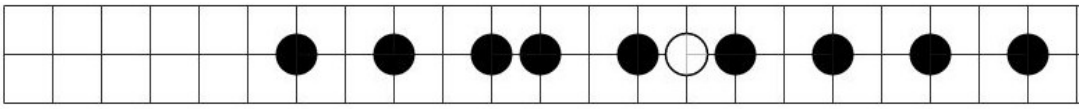
Alternativa, ki ne pripelje *Levega* do zmage (*Levi* doda kamen):



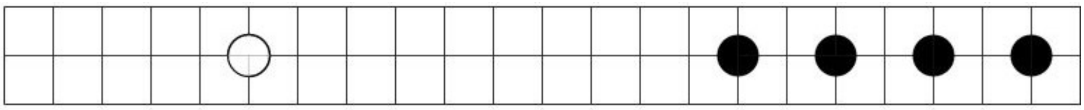
D2 L1



D3 D1 L2 L4



D2 L1 L3

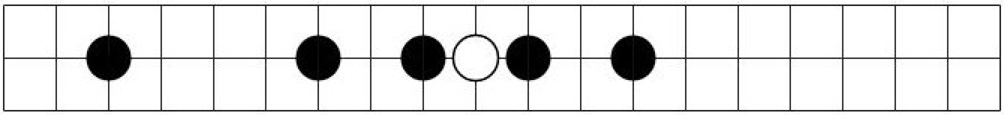


To je zmagovalna pozicija *Desnega*. Premislek prepuščamo bralcu.

3.2 Lihi enorazsežni figomet

Zdi se, da je popolna zmagovalna strategija za enorazsežni *figomet* onstran našega dosega. Vendar, če dodamo majhno omejitev na dovoljene poteze, lahko pridelamo popoln opis zmagovalne strategije.

Lihi figomet je enorazsežni *figomet*, v katerem lahko igralci dodajo kamen na prazen prostor, ki je liho oddaljen od žoge. Tako se znebimo vseh praznih kvadratkov, ki so sodo oddaljeni od žoge. Primer pozicije *lihega figometa*.



V tej igri z omejitvami, bi lahko podali hipotezo, da bo optimalni igralec skočil samo takrat (skok imenujemo *skok na zmagovito pozicijo*), ko bo ta skok dovolj blizu nasprotnikovega gola ter se ne more več vrniti nazaj na prvotno mesto. Izkaže se, da je ta hipoteza pravilna. Za vsakega igralca lahko definiramo potencialno funkcijo, koliko kamnov mora dodati preden lahko skoči na *zmagovito pozicijo*. Opisali bomo kako izračunamo to *potencialno funkcijo* in pokazali, da ima igralec z nižjim potencialom vedno možnost zmage.

Leva potencialna funkcija upošteva samo kamen desno od žoge, *desna potencialna funkcija* upošteva samo kamne levo od žoge. *Potencialni funkciji* sta definirani simetrično, opisujemo zgolj levo.

Najprej definiramo pomožno količino, imenujemo jo *seštevek* (levi oziroma desni). Začetna vrednost levega *seštevka* je 0, seštevati začnemo na skrajnem desnem robu igrišča in se korakoma bližamo žogi ter upoštevamo samo liha polja.

i. Če je (lihi) prostor nezaseden *seštevku* prištej 1,

ii. če je na (lihem) prostoru kamen, potem od *seštevka* odštej 1, razen če bi s tem *seštevek* postal negativen; v tem primeru naj ostane enak 0.

Končni *levi seštevek*, ki ga dobimo, če upoštevamo vsa liha mesta desno od žoge, označimo z $bsum$, *levi seštevek*, pri katerem ne upoštevamo lihega mesta neposredno desno od žoge pa z $psum$.

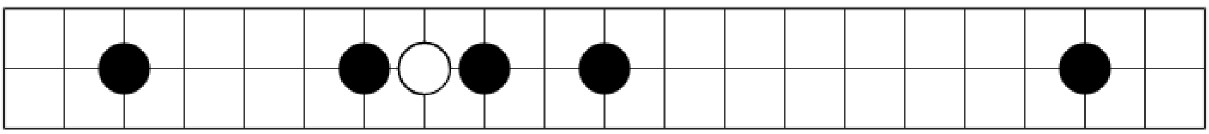
Levo potencialno funkcijo p_L definiramo kot

$$p_L = \lfloor bsum/2 \rfloor + (1 - \delta_{psum}^{bsum}),$$

kjer je δ_{psum}^{bsum} Kroneckerjev delta, ki je enak 1, če je *bsum* = *psum*, oziroma 0 v nasprotnem primeru.

Opazimo lahko, da je δ_{psum}^{bsum} = 1 samo v primeru, ko je *bsum* = *psum* = 0. Če je namreč *psum* > 0, potem se *bsum* in *psum* razlikujeta za natanko 1, saj *bsum* iz vrednosti *psum* izračunamo tako, da 1 bodisi odštejemo bodisi prištejemo.

Primer:



Za *Levega* je tako *psum* = 2 (1, 0, 1, 2, 3, 2) in *bsum* = 1 (1, 0, 1, 2, 3, 2, 1). V oklepajih je *seštevek* po korakih. Tako je $p_L = 1$.

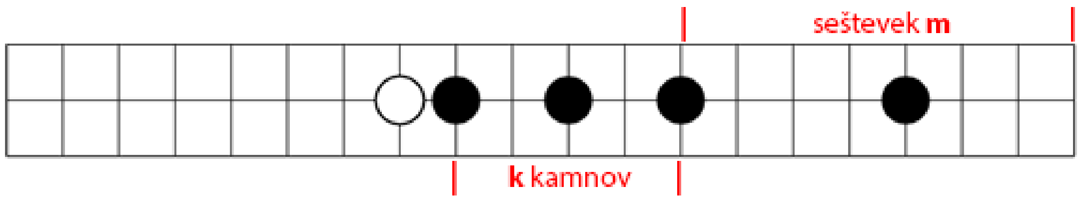
Izrek 1. Za pozicijo S *lihega figometa* velja

- Če je $p_L < p_D$, potem je S zmagovalna pozicija za *Levega*,
- če je $p_D < p_L$, potem je S zmagovalna pozicija *Desnega*,
- če je $p_L = p_D$, potem je S zmagovalna pozicija za tistega igralca, ki je na potezi.

Lema 2. V poziciji *lihega figometa*:

1. Če je $p_L = 0$, potem lahko *Levi* skoči na pozicijo, za katero velja $p_L < p_D$.
2. Če je $p_L > 0$, potem lahko *Levi* igralec zmanjša vrednost p_L za 1, tako da doda kamen na desno stran od žoge.
3. Če je $p_L > 0$, potem vsak skok *Levega* poveča $p_D - p_L$ za največ 1.

Dokaz. Če so kamni samo na desni strani od žoge potem je $p_L = 0$ in *Levi* lahko skoči in takoj zmaga. Opazujemo torej pozicijo, kjer neposredno desno od žoge stoji k zaporednih kamnov (pazi, kamni so lahko postavljeni zgolj na lihih mestih), na naslednjem lihem mesta pa kamna ni. Prav tako naj m označuje vrednost (lihega) *seštevka* neposredno preden začnemo upoštevati skušino teh k kamnov.



1. Če je $p_L = 0$, potem je $0 < m < k$ in posebej $k > 1$. Izberimo tako sodo število k', da je $k - 1 \leq k' \leq k$. Zato je $k' > 0$ in $(k' - 1)/2 + 1 = k'/2$. Potem *Levi* spremeni p_L iz 0 na največ $(k' - 1)/2 + 1$ tj. $k'/2$, če skoči preko k' kamnov. Sedaj se p_D spremeni iz $m'/2 + 1$ v $(m' + k')/2 + 1$ za $m' > 0$ ali iz 0 v $k'/2 + 1$. V obeh primerih velja za novo pozicijo $p_L < p_D$
2. Če je $p_L > 0$, potem je $m \geq k$. Če bi namreč veljalo $k > m$, potem bi za *potencialno funkcijo* p_L pridelali 0 – z vsakim od k kamnov zmanjšamo vrednost *seštevka* in tako *psum* kot *bsum* sta enaka 0.
Če dodamo kamen neposredno za skupino k kamnov na desno, potem *potencialno funkcijo* p_L strogo povečamo – če je $m = k$ ali $m = k + 1$, potem p_L zmanjšamo iz 1 na 0, sicer pa iz $\frac{m-k}{2} + 1$ na $\frac{m-k}{2}$.

3. Predpostavimo, da je $p_L = (i/2) + 1 > 0$, pri nekem $i \geq 0$. Naj *Levi* skoči naprej preko j kamnov. Potencial *Levega* se spremeni iz $(i/2) + 1$ v $((i + j)/2) + 1$. Potencial *Desnega* se spremeni bodisi iz $r/2 + 1$ v $(r + j)/2 + 1$ pri nekem $r \geq 0$, bodisi iz 0 v $(j/2) + 1$. V obeh primerih je sprememba potenciala med $j/2$ in $j/2 + 1$, torej se $p_D - p_L$ spremeni za največ 1.

Dokaz izreka 1. Pokazati je potrebno, da *Levi* ne more izgubiti ter, da je igra končna. V tem primeru mora *Levi* zmagati.

Predpostavimo, da je $p_L < p_D$ in *Desni* na potezi. Opomnimo, da *Desni* ne more zmagati v eni potezi, saj je $p_D > 0$.

Če *Desni* doda kamen na levo stran od žoge, potem zmanjša potencial za največ 1 in ne vpliva na potencial *Levega*. Če *Desni* doda kamen na desno stran od žoge ne vpliva na p_D, ampak lahko zmanjša le p_L. Če *Desni* skoči naprej ali nazaj se $p_L - p_D$ spremeni za največ 1. V vseh primerih je na potezi *Levi* in velja $p_L \leq p_D$.

Privzemimo zdaj, da velja $p_L = 0$ in naj k označuje število kamnov v nizu desno od žoge. *Levi* skoči preko k' kamnov, kjer je $k' = k$ ali $k' = k - 1$ in je k' sodo. S takšnim k' imamo $\lfloor (k' - 1)/2 \rfloor + 1 = \lfloor k'/2 \rfloor$. Nato po Lemi 2.1 velja $p_L < p_D$ in je na vrsti *Desni*. Če je $p_L > 0$ *Levi* doda kamen na konec sosednjega niza kamnov in se p_L zmanjša za najmanj 1 ter po Lemi 2.2 velja $p_L < p_D$ in je na vrsti *Desni*.

Na koncu predpostavimo, da je na vrsti *Levi* in velja $p_L \leq p_D$. Če je $p_L = 0$, potem po Lemi 2.1 lahko *Levi* skoči na pozicijo, za katero velja $p_L - p_D \geq 1$. Če je $p_L > 0$ po Lemi 2.2 lahko Levi doda kamen, da zmanjša p_L za 1, tako da ne spremeni p_D. V obeh primerih je na vrsti *Desni* z $p_L < p_D$.

Sledi, da če je na vrsti *Levi* in velja $p_D \geq p_L$ ali *Desni* in je $p_D > p_L$, potem *Desni* ne more zmagati, saj ni nikoli na vrsti, ko je $p_D = 0$ in tako ne more skočiti.

Na koncu moramo dokazati, da se igra zaključi po končno mnogo potezah. Denimo, da to ni res in, da igro igramo neskončno mnogo potez. Pri tem nekatera igralna polja neskončno mnogokrat spremenijo status – iz praznega na zasedenega s kamnom in nazaj. Z A označimo najbolj desno igralno polje, ki status menja neskončnokrat.

Privzemimo še, da je igra napredovala tako daleč, da ni več odigranih potez desno od A. Ker v igri tudi v nadaljevanju spreminjamo status polja A, opazujmo potezo, ki žogo spravi desno od polja A.

Je lahko na potezi levi igralec? Ne, saj bi moral v skladu s svojo strategijo postaviti kamen desno od žoge in s tem od A (in bi spremenil status preveč desnemu polju) ali pa doseči zadetek (kar je v nasprotju z neskončno dolžino igre). Če pa je na potezi *Desni*, potem je *Levi* ravnokar skočil desno od A. *Desni* ne more skočiti v smeri nasprotnikovega gola (saj neposredno levo od žoge ni kamna), zato ga doda, ravno tako pa ga v naslednji potezi doda *Levi* (in sicer desno od žoge). To je znova v nasprotju s spremembami statusa polj desno od A.

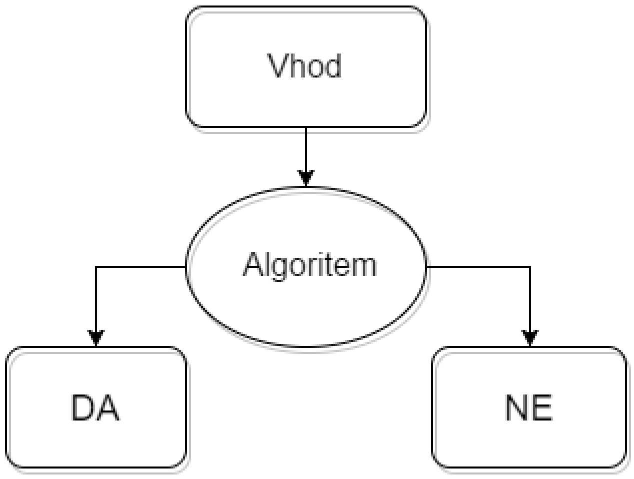
Torej je igra končna in, ker *Desni* ne more zmagati, je zmagovalec *Levi*.

Poglavje 4 NP polnost končnice

V *šahovski* partiji je relativno enostavno videti, ali lahko v eni sami potezi matiramo nasprotnika. Takšni potezi pravimo *zmagovita poteza*. Za razliko od *šaha*, je vprašanje, ali ima igralec *figometa* zmagovito potezo, težavno. V tem poglavju predstavimo rezultat, da je v igri *figometa* v splošnem težko odločiti, ali lahko nasprotnika premagamo z eno samo skakalno potezo. Ta problem imenujemo *problem končnice figometa.*

4.1 NP-polnost

Odločitveni problem je vprašanje o pripadnosti strukture neki množici. Imamo torej vprašanje, za katerega imamo poljuben vhod in samo dva izhoda oziroma odgovora: DA ali NE.



Rezultat odločitvenega problema lahko predstavimo kot podmnožico za katero velja odgovor DA. Tako dobimo enostaven zapis: če $x \in Q$, potem velja $Q(x) = DA$ in, če $x \notin Q$, potem velja $Q(x) = NE$.

Teorija računske zahtevnosti se ukvarja s klasifikacijo odločitvenih (tudi optimizacijskih) problemov glede na težavnost. Za razumevanje našega primera so pomembni razredi:

- **P**. Problemi, ki so rešljivi v polinomskem času. Obstaja učinkovit (polinomske časovne zahtevnosti) algoritem, da poiščemo rešitev.

- **NP**. Problemi, ki so rešljivi v nedeterminističnem polinomskem času. Obstaja učinkovit algoritem za preverjanje pravilnosti rešitve.

- **PSPACE**. Problemi, ki jih lahko rešimo, če imamo na voljo polinomsko mnogo prostora, ne glede na to, kako dolgo porabimo za takšno rešitev.
- **EXPTIME**. Problemi, ki so rešljivi v eksponentnem času.

Za zgoraj omenjen razrede veljajo naslednje zveze: P ⊆ NP ⊆ PSPACE ⊆ EXPTIME.

Odločitveni problem je NP-težak, če je vsaj tako težak, kot so najtežji problemi razreda NP. Odločitveni problem je NP-poln, če je NP-težak in pripada razredu NP. V praksi za problem X pokažemo, da je NP-poln, takole: najprej moramo preveriti, da sploh pripada razredu NP, nato pa znan NP-poln problem Y s polinomsko prevedbo prevedemo na X. Pišemo tudi Y ≤p X.

Znano je, daje problem 3-izpolnjivosti NP-poln 0. Odločitveni problem 3-izpolnjivosti, imenovan tudi 3-SAT, lahko opišemo takole. Vhodni podatek problema 3-SAT je izjavni izraz v 3-CNF, konjunktni normalni obliki, v kateri ima vsak člen (osnovna disjunkcija) tri literale (spremenljivke oziroma njihove negacije). Zgled takšnega izjavnega izraza je:

$I = (p \vee q \vee \neg r) \wedge (\neg p \vee r \vee s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (q \vee r \vee \neg s)$

Ta izjavni izrez vsebuje štiri člene, vsak izmed členov je disjunkcija treh spremenljivk oziroma njihovih negacij. V tem izrazu nastopajo štiri spremenljivke p, q, r, s. Na izhodu problema 3-SAT pričakujemo odločitveni odgovor, da ali ne, ali lahko izberemo nabor logičnih vrednosti spremenljivk p, q, r, s, pri katerem ima celoten izraz I logično vrednost 1. V takšnem naboru bodo vsi členi izraza I, $(p \vee q \vee \neg r)$ *in* $(\neg p \vee r \vee s)$ *in* $(\neg p \vee \neg q \vee \neg s)$ *in* $(q \vee r \vee \neg s)$ imeli logično vrednost 1, saj so vezani konjunktivno. Še več, za vsakega od členov izraza I je potrebno določiti razlog, zakaj je izpolnjen – vsaj eden od treh njegovih literalov mora imeti logično vrednost 1.

Za izjavni izraz I velja, da je izpolnjiv, saj je denimo nabor logičnih vrednosti p = 0, q = 1, r = 0 in s = 1 njegov model – nabor pri katerem je I resničen. V tem naboru je tako resničen vsak člen izraza I, v njegovem prvem členu $(p \vee q \vee \neg r)$, denimo, sta resnična kar dva literala, tako q kot $\neg$ r.

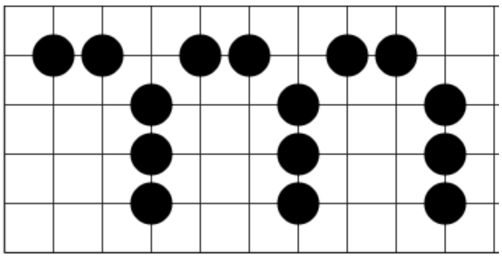
4.2 *Figomet* in NP-polnost

V tem razdelku bomo pokazali naslednji izrek.

Izrek 2. *Končnica figometa* je NP-polni problem.

Kako pokažemo, da je odločitveni problem NP-poln? Kot prvo je potrebno premisliti, da problem pripada razredu NP, kot drugo pa je dovolj konstruirati prevedbo na katerega od znanih NP-polnih problemov. Kot smo v prejšnjem razdelku namignili bomo problem končnice prevedli na 3-SAT.

Za začetek si bomo ogledali tri konfiguracije S_1, S_2 in S_3 kamnov, ki jih bomo potrebovali v konstrukciji prevedbe v dokazu Izreka 2. V vseh treh konfiguracijah si predstavljamo, da žoga stoji neposredno levo od najbolj levega prikazanega kamna.

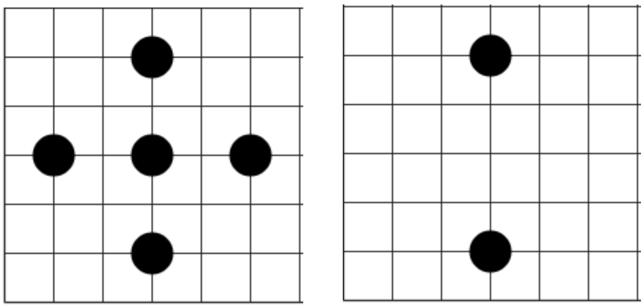


Slika 4.1: Situacija S_1.

Na zgornji sliki je prikazana situacija S_1. Opazimo, da lahko žoga, ki priskače z leve strani svojo pot nadaljuje vzdolž katerega koli od vertikalnih repkov. Če preskoči zgolj prvi par točk v horizontali, lahko nadaljuje vzdolž prve vertikale, če pa, denimo, preskoči vse tri pare točk v horizontalni smeri, pa lahko nadaljuje skok vzdolž zadnje vertikale. Situacijo S_1 lahko posplošimo tudi na večje število vertikalnih repkov, ravno tako bi število vertikalnih repkov lahko zmanjšali na dva ali celo na en sam vertikalni rep. Situacijo S_1 si lahko predstavljamo kot vejitev skoka žoge, ki se iz predpisanega začetka lahko nadaljuje v več možnih smeri. Po drugi strani pa lahko situacijo S_1 razumemo v nasprotni smeri skoka. Če žoga priskače od spodaj vzdolž katere koli vertikalne trojice točk, lahko skok nadaljuje v horizontalni smeri proti levi. Konfiguracijo S_1 lahko torej smatramo tudi za spoj, ki več možnih vhodov napelje v en sam izhod.

Situaciji S_2 in S_3 sta simetrični, prikazani pa sta v dveh razporeditvah: neposredno pred oziroma neposredno po skoku z žogo.

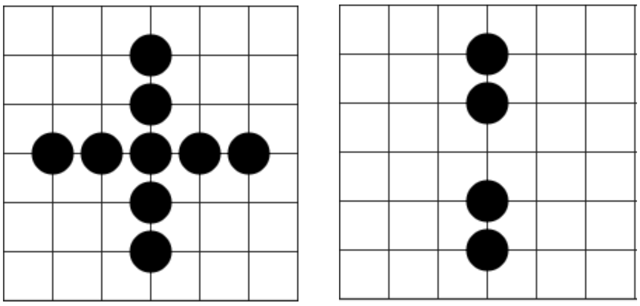
Oglejmo si najprej situacijo S_2.



Slika 4.2: Situacija S_2.

Če z žogo začnemo na levem polju (neposredno desno od njega je kamen), potem lahko skok s žogo nadaljujemo proti desni. Če preskočimo tri kamne, pridelamo situacijo na sliki desno. Pri tem pa dobljena situacija ne omogoča vertikalnega skoka z žogo, če predhodno ne dodamo kakšnega kamna. Posebej, v isti potezi čez tako križno konfiguracijo S_2 **ne moremo** skočiti v vertikalni smeri, če smo poprej z žogo skočili v horizontalni smeri.

Situacija na sliki S_3 je znova simetrična, znova predpostavimo, da je žoga na začetku neposredno levo od najbolj levega prikazanega kamna.



Slika 4.3: Situacija S_3.

V tem primeru pa horizontalen skok žoge vseeno dopusti, da v **isti** potezi skok z žogo nadaljujemo še v vertikalni smeri (če preostali kamni v celotni poziciji dopustijo preskok žoge na ustrezno mesto). Horizontalen skok z žogo nam po odstranjevanju petih zaporednih kamnov prihrani dve vertikalni skupinici po dveh kamnov, vzdolž katerih bi lahko s kamnom skočili še drugič preko istega polja.

Na tem mestu omenimo, da lahko konfiguraciji S_2 in S_3 uporabimo tudi v primeru, ko z žogo skačemo z leve, in celo v primeru, ko žoga skače v vertikalni smeri. Ravno tako bomo S_1, ustrezno zavrteno ali prezrcaljeno seveda, lahko uporabili z žogo iz desne ali pa iz vertikalne smeri.

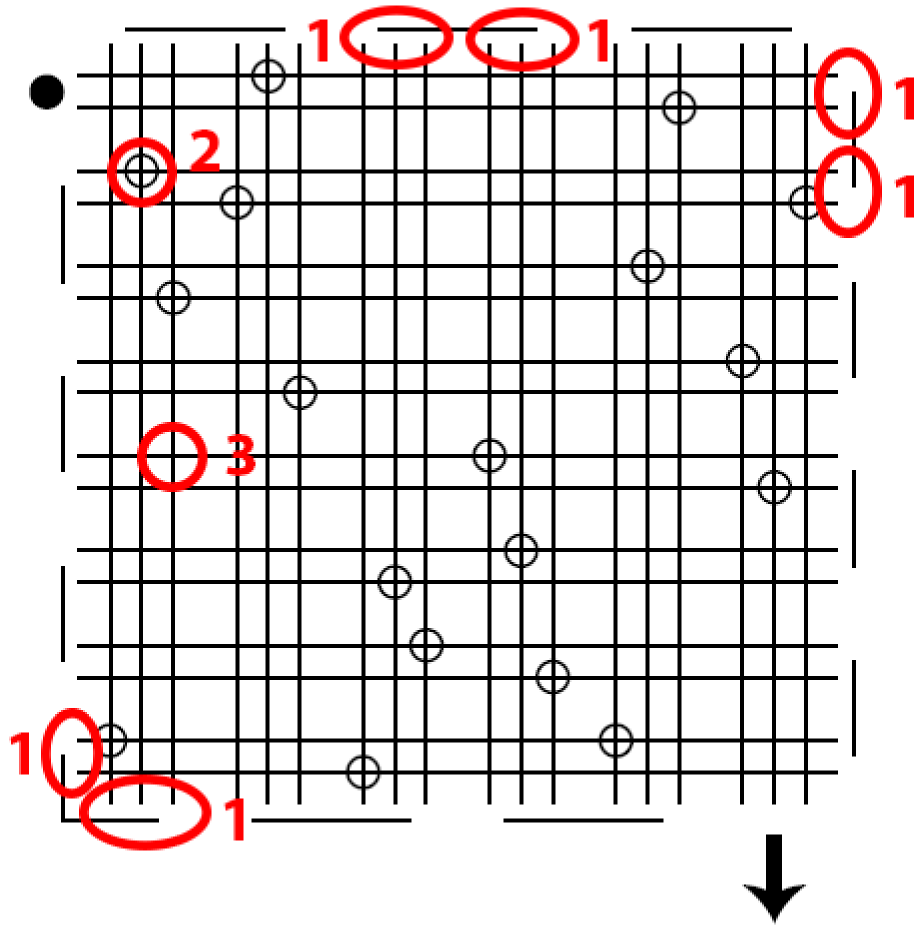
Pripadnost razredu NP. Potezo, zaporedje skokov, lahko opišemo z zaporedjem puščic, denimo gor, dol, levo,…V vsaki dopustni potezi imamo največ $8n^2$ skokov, saj lahko na posamezno igralno polje v potezi skočimo največ osemkrat – nikoli namreč posameznega polja ne moremo dvakrat doseči v isti smeri. To pomeni, da bo vsaka dopustna poteza dolžine največ $8n^2$. Če igralno površino predstavimo kot n x n matriko s tremi možnimi koeficienti - prazno polje, kamen, žoga, je dolžina zmagovalne poteze polinomsko omejena z velikostjo vhodnih podatkov. Brez težav lahko v polinomskem času preverimo, ali podana poteza pripelje žogo v gol – enostavno jo moramo modelirati.

Prevedba na problem 3-SAT. Za vsak izjavni izraz J v 3-CNF obliki je potrebno zgraditi igralno situacijo *figometa* S_J, za katero velja, da ima J model natanko tedaj, ko lahko igralec *figometa* v poziciji S_J zabije gol v eni potezi. Prevedbo *figometa* na 3-SAT bomo opisali na dovolj splošnem zgledu.

Izberimo izjavni izraz:

$J = (\neg p_2 \vee p_3 \vee \neg p_8) \wedge (\neg p_1 \vee p_2 \vee p_4) \wedge (p_6 \vee \neg p_7 \vee p_8) \wedge (\neg p_5 \vee p_6 \vee p_7) \wedge (p_1 \vee \neg p_3 \vee \neg p_8) \wedge (p_2 \vee \neg p_4 \vee p_5)$

S šestimi členi $C_1,\ldots,C_6$ in osmimi spremenljivkami $p_1,\ldots,p_8$. zgradimo igralno situacijo S_J takole: spremenljivkam v izrazu J bodo ustrezale horizontalne verige kamnov, členom v izrazu J pa vertikalne verige kamnov.



Slika 4.4: Igralna situacija S_J [1]. Oznake 1, 2, 3 so zgoraj opisane situacije S_1, S_2, S_3.

Vsak literal *x* v členu je bodisi spremenljivka bodisi njena negacija. Literalu *x* v posameznem členu ustreza natanko ena vertikala, ki se seka s horizontalami, ki predstavljajo spremenljivke. Če je *q* spremenljivka, ki nastopa v literalu *x* (tj. *x* je bodisi enak *q* bodisi negaciji spremenljivke *q*), bomo natančno eno od križišč vertikale *x* s horizontalama spremenljivke *q* z odstranjevanjem petih kamnov pripeljali v situacijo S_2.

Vsako spremenljivko p_i predstavimo s paroma vzporednih horizontalnih vrstic kamnov na koordinatah $20i$ in $20i + 5$. Pri tem se dogovorimo, da horizontalna vrstica $20i$ ustreza logični vrednosti $p_i = 0$, medtem ko vrstica $20i + 5$ ustreza logični vrednosti $p_i = 1$. Člen C_j predstavimo s trojico vzporednih vertikal kamnov na koordinatah $20j$, $20j + 5$, $20j + 10$. Pri tem vertikala $20i$ ustreza prvemu, $20j + 5$ drugemu in $20j + 10$ tretjemu literalu.

Če je denimo spremenljivka p_i tretji literal v členu C_j, potem na križišču horizontale $20i$ in vertikale $20j + 10$ odstranimo pet kamnov in lokalno pridelamo situacijo S_2, medtem ko na križišču horizontale $20i+5$ in vertikale $20j + 10$ ohranimo situacijo S_3. Če pa je, denimo, negacija spremenljivke p_k drugi literal v členu C_j, potem na križišču horizontale $20k + 5$ in vertikale $20j + 5$ odstranimo štiri kamne in pridelamo situacijo S_2, na križišču horizontale $20k$ in vertikale $20j + 5$ ohranimo situacijo S_3.

Zaporedne pare horizontal in zaporedne trojice vertikal povežemo s situacijami S_1. Ravno tako s situacijami S_1 omogočimo, da žoga, na sliki Slika 4.44.1 je prikazana levo zgoraj, začne skok vzdolž ene od zgornjih dveh horizontal. Nazadnje pa zadnjo trojico vertikal, na sliki Slika 4.44.1 so prikazane skrajno desno, podaljšamo do gola.

Žogo nastavimo na začetek horizontal, ki ustrezata spremenljivki p_1, zmagovalna poteza pa naj žogo pripelje v gol, ki je na sliki predstavljen v smeri puščice. Zmagovalni skok žoge mora za vsako od spremenljivk izbrati eno od horizontal, bodisi zgornjo, kjer ima spremenljivka logično vrednost 0, ali spodnjo, ki predstavlja resnično vrednost spremenljivke. Poteza vzdolž parov horizontal spremenljivk poteka izmenjaje levo desno, premik navzdol med dvema zaporednima paroma horizontal omogočimo z dvema situacijama S_1. Ko zaključimo s spremenljivkami, žoga nadaljuje skoke v vertikalni smeri izmenjaje gor in dol. Za vsak člen izbere enega od literalov, ki je v tem členu izpolnjen, eno od treh vertikal, ki na svoji poti sreča natanko eno situacijo 2. Toda, pravilen literal pomeni, da žoga v prvem delu skoka ni uporabila tiste horizontale, ki se v situaciji S_2 seka s trenutno vertikalo. Pot žoge vzdolž vertikal torej izbira literale v členih izraza J, ki so resnični. Zato žoga, v primeru, ko doseže gol, opiše izbiro logičnih vrednosti modela izraza J. In obratno, če žoge ne moremo pospraviti v gol v eni sami skakalni potezi, potem ne moremo izbrati modela za izjavni izraz J.

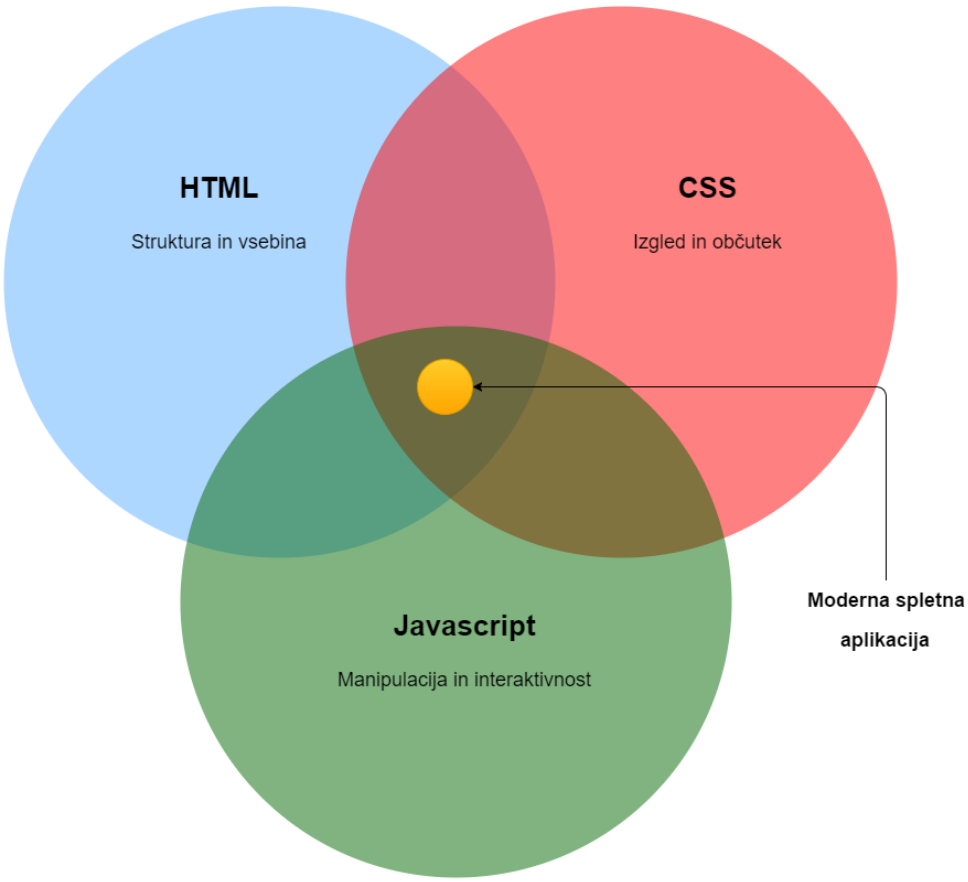
Zmagovalna poteza torej v delu poti levo-desno izbere logične vrednosti spremenljivke $p_1,...p_8$, pri katerih so vsi členi $C_1,...C_6$ zaradi vertikalnega dela poteze izpolnjeni. Gol torej lahko zabijemo natanko tedaj, ko ima izjavni izraz J model.

Poglavje 5 Implementacija igre za dva igralca

Igranje igre je najlažji način, da jo spoznamo, zato bomo implementirali aplikacijo za dva igralca. Razviti želimo aplikacijo, ki bo delovala na računalnikih, tabličnih računalnikih in pametnih telefonih. Uporabili bomo tehnologije, primerne za spletne brskalnike, ker želimo, da je igra dostopna čim širšemu krogu uporabnikom. Za razvoj interaktivne igre pa potrebujemo tehnologijo, ki omogoča prav to.

5.1 Uporabljene tehnologije

Potrebovali smo HTML, CSS in Javascript, da smo lahko razvili moderno spletno aplikacijo.



Slika 5.1: Potrebne tehnologije za razvoj aplikacije

HTML je označevalni jezik, ki ga uporabljamo za ustvarjanje spletnih strani in aplikacij. Opisuje semantično strukturo spletne strani s pomočjo elementov za naslov, odstavek, seznam, povezavo, sliko, obrazec,... Ti elementi so označeni s pomočjo lomljenih oklepajev, ki jih imenujemo značke. Razlikujemo med takimi, ki neposredno vstavljajo vsebino (primer: `<input />`, `<img />`), in tistimi, ki obdajajo besedilo in lahko znotraj vsebujejo tudi druge značke (primer: `<h1></h1>`, `<p>…</p>`). V naši aplikaciji uporabljamo zadnjo verzijo imenovano HTML5, ki je bila izdana oktobra leta 2014. Ta verzija je izboljšana predvsem pri podpori za moderne večpredstavnostne medije in ohranja berljivost za človeka ter računalnike in ostale naprave.

Primer strukture v naši aplikaciji:

```
<div role="tabpanel" class="tab-pane" id="profile">
    <br>
    <h4><span class="glyphicon glyphicon-play" aria-hidden="true"></span> Introduction</h4>
        <p>It's a simple game. Played on 15x19 board.<br>
        There are only two types of pieces:</p>
        <ul>
            <li><strong>Ball</strong>: white circle</li>
            <li><strong>Players</strong>: black filled circle</li>
        </ul>

    <br>
    <h4><span class="glyphicon glyphicon-move" aria-hidden="true"></span> Moves</h4>
```

CSS je jezik, s katerim opišemo predstavitveni del dokumenta, ki smo ga definirali s HTML. Tako predvsem določimo vizualni izgled spletne aplikacije. Ker je po navadi datoteka CSS ločena, lahko izgled preprosto spremenimo, ne da posegamo v HTML kodo. Za CSS velja tudi pravilo dedovanja stilov. Sintaksa je zgrajena iz selektorja in eno ali več lastnostmi z željenimi vrednostmi. Primer:

```
#players div {
    width:100px;
    height:30px;
    text-align: center;
    font-size:16px;
    padding:5px;
}
#players div.active {
    background-color: #337ab7;
    color: #fff;
}
```

Uporabili smo različico CSS3, ki jo podpira večina modernih spletnih brskalnikov in omogoča nekaj novih naprednih modulov.

Javascript je visokonivojski skriptni jezik, ki ga je razvil Netscape, da bi spletnim programerjem pomagal pri ustvarjanju interaktivnih spletnih strani. Izvajanje poteka na brskalniku, preko katerega pa dostopa uporabnik, zato lahko prihaja do minimalnega odstopanja v končnem rezultatu. Podpirajo pa ga vsi novejši spletni brskalniki. Strukturna programska sintaksa je v veliki meri povzeta po programskem jeziku C. Ponovno smo uporabili zadnjo stabilno različico ECMAScript 6, predvsem zaradi podpore za deklariranje razredov. Primer je spodaj:

```
class Phutball {
    constructor(grid,options) {
        this.grid = grid;
        this.options = options;

        this.addGoals();
    }

returnDirections(direction) {
    var directionsObject = {
        'N': {
            'j': -1,
            'i': 0
        },
```

5.2 Knjižnice

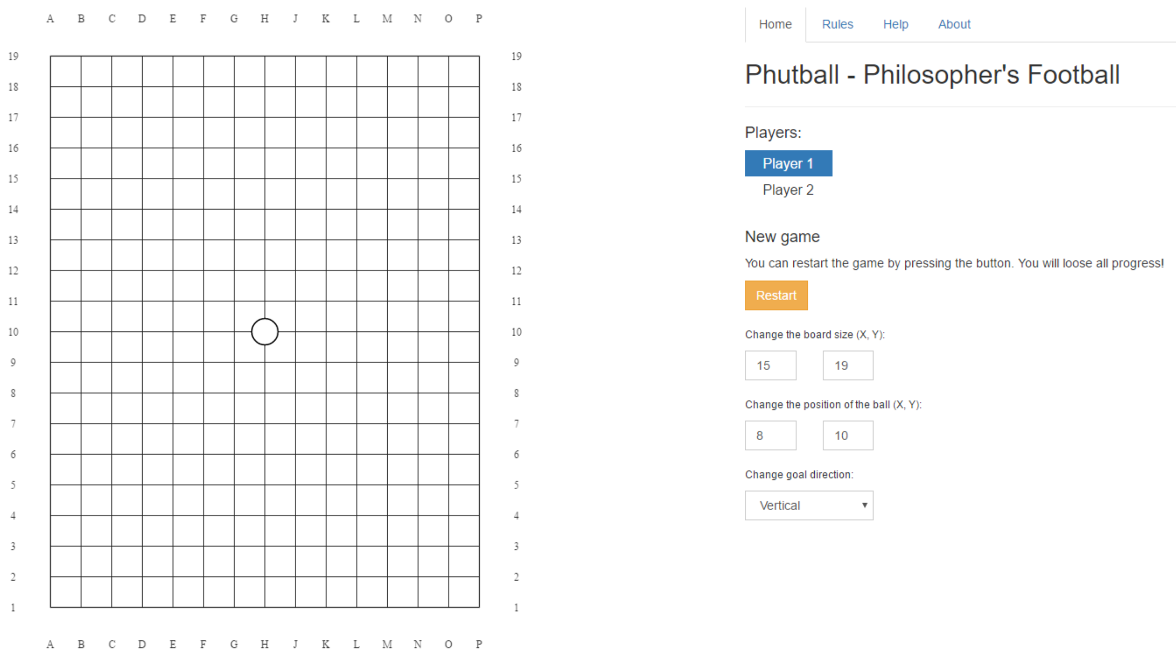
Za splošen moderen izgled smo uporabili **Bootstrap** [11], ki sledi modernim smernicam in nam omogoča, da imamo hitro postavljen vizualni del aplikacije. Je ena izmed najbolj popularnih HTML, CSS in Javascript knjižic. Uporabili smo predvsem implementacijo za tipografijo, zavihke, gumbe in prilagodljivost za mobilne naprave.

Implementacijo interaktivnosti lahko tudi nekoliko poenostavimo, če si pomagamo z močno razširjeno knjižico za Javascript imenovano **Jquery** [9]. Pomaga predvsem pri izbiranju elementov v HTML strukturi, ustvarjanju animacij, upravljanje z akcijami ter razvijanje AJAX aplikacij.

Za prikaz igralne plošče in izvajanje potez smo poiskali primerno knjižico. Imenuje se **JGoBoard** [12] in je primarno narejena za igro Go. Vendar ima možnost določanja velikosti igralne plošče, ter funkcije za izvajanje akcij ob premiku miške na določeno polje in njen klik. Narejena je tudi z modernim HTML5 elementom *canvas*, tako da deluje tudi na mobilnih napravah, saj ne obremenjuje preveč sistemskih virov naprave.

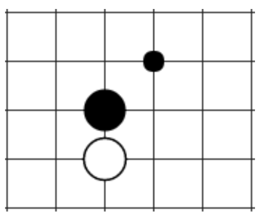
5.3 Grafični uporabniški vmesnik

Na začetku imamo pred sabo na levi strani igralno ploščo velikosti 15x19 in zavihke *Domov*, *Pravila*, *Pomoč* in *O aplikaciji* na desni strani. Zavihek *Domov* nam pove, z modrim ozadjem, kateri igralec je na potezi. Zraven imamo tudi gumb za ponastavitev igre.

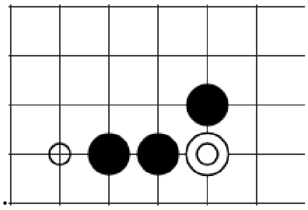


Slika 5.2: Uporabniški vmesnik

Kamen dodamo tako, tako da se z miško premaknemo na željeno mesto. Če je mesto nezasedeno, se nam prikaže majhen črn poln krog, na katerega lahko kliknemo, pri čemer se izriše velik poln črni krog, ki predstavlja kamen.

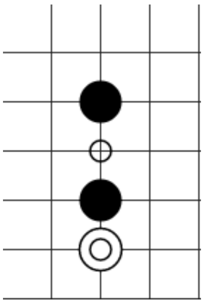


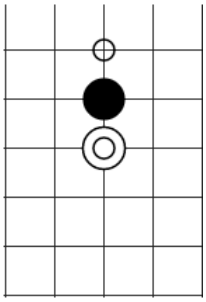
Če želimo izvesti skok, se z miško pomaknemo na žogo, ki je predstavljena z belim krogom. Ko kliknemo nanjo, se znotraj žoge pojavi majhen krog, kar pomeni da smo aktivirali fazo skoka. Sedaj lahko izvedemo skok, tako da se pomaknemo z miško na željeno mesto. Če je skok dovoljen, skačemo lahko v vse 8 smeri preko več kamnov, se pojavi na tem mestu majhen krog



V primeru, da smo s tem skokom zadovoljni, kliknemo in skok se izvede. Žoga skoči na to mesto, ter hkrati odstrani vse kamne, ki smo jih preskočili. Skoki izven igralne plošče na levi in desni niso dovoljeni.

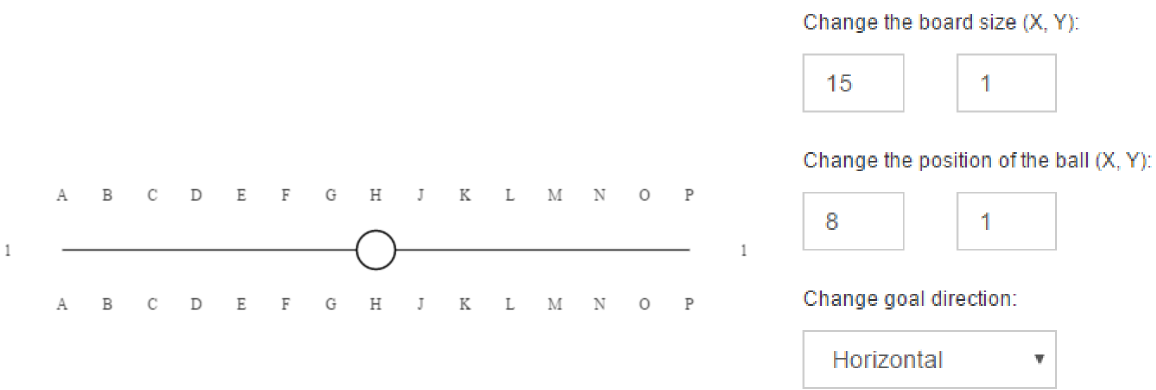
Skočimo lahko tudi večkrat zapored, če je skok mogoč.





Cilj igre je doseči zadetek. To dosežemo tako, da spravimo žogo na ali preko nasprotnikovega gola. Gol prvega igralca je zgornja vrstica igralne površine, gol drugega pa čisto spodnja vrstica.

V tretjem poglavju smo predstavili enorazsežni *figomet*, ki ga tudi lahko igramo s pomočjo naše aplikacije. Na desni strani imamo vnosna polja, kjer lahko nastavljamo dolžino in širino igralnega polja. Prav tako nastavimo začetno pozicijo žoge in smer golov. Spodaj je primer nastavitev za igranje enorazsežnega *figometa*.



Poglavje 6 Zaključek

Povzeli smo nekaj značilnosti kombinatoričnih iger, ki so pomembne za računsko zahtevnost. Predstavili smo, zakaj je kombinatorična igra *figomet* težka, in dokazali, da je odločiti, ali lahko igralec v dani situaciji zmaga z eno potezo, NP-poln problem.

Igro *figometa* za dva igralca, ki smo jo razvili za spletne brskalnike, lahko igramo na poljubnih dimenzijah, saj smo dodali vnosna polja, preko katerih lahko določamo dolžino in širino igralne plošče. Tako lahko uporabniki tudi sami preverijo igralne poteze predstavljene v enorazsežnem *figometu*. Aplikacijo bi lahko nadgradili tako, da bi lahko igrali proti računalniku. Ker je igra težka, bi optimalnega računalniškega igralca težko sprogramirali. Celo v *enorazsežnem figometu* so lahko zmagovalne poteze nepričakovane. Ali obratno, nekatere na prvi pogled smiselne poteze vodijo v poraz. Zato implementacijo računalniškega igralca *figometa* zapuščamo kot izziv prihodnjim generacijam.

Literatura

[1] G. Alexander, *Play Chess*. [Online]. Dosegljivo: http://gamejolt.com/games/play-chess/164040

[2] E. Berlekamp; J. H. Conway; R. Guy. *Winning Ways for your Mathematical Plays*. I. Academic Press, 1982.

[3] S. Chinchalkar, "An Upper Bound for the Number of Reachable Positions". *ICCA Journal, Vol. 19, No. 3*, str. 181-183, 1996.

[4] E. D. Demaine, M. L. Demaine, D. Eppstein, "Phutball Endgames are Hard", v *More Games of No Chance*, MSRI Publications 42, Cambridge Univ. Press., str. 351-360, 2002.

[5] A. Fraenkel, D. Lichtenstein, "Computing a perfect strategy for n×n chess requires time exponential in n", *J. Comb. Th. A (31)*, str. 199-214, 1981.

[6] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., NY, ZDA, 1979.

[7] J. P. Grossman, R. J. Nowakowski, "One-Dimensional Phutball", v *More Games of No Chance*, MSRI Publications 42, Cambridge Univ. Press., str. 361-367, 2002.

[8] P. M. Grundy, "Mathematics and games", *Eureka*, The Archimedeans, str. 6-8, 1939.

[9] JQuery Foundation, *Jquery*.[Online]. Dosegljivo: https://jquery.com.

[10] C. Marie, *Jeux de société japonais sur internet*. [Online]. Dosegljivo: http://www.kelbet.com/jeux-de-societe-japonais-sur-internet

[11] M. Otto, J. Thornton, *Bootstrap: Designed for everyone, everywhere*. [Online]. Dosegljivo: http://getbootstrap.com.

[12] J. Pihlajamaa, *JGoBoard.* [Online]. Dosegljivo: http://jgoboard.com

[13] R. Scharnagl, *Chess and numbers*, 2001. [Online]. Dosegljivo: https://web.archive.org/web/20070613072827/http://www.chessbox.de/Compu/schachzahl2_e.html

[14] R. P. Sprague, »Über mathematische Kampfspiele«, *Tohoku Mathematical Journal*, vol. 41, str. 438–444, 1935.

[15] Wikipedia, *English draughts*,. [Online]. Dosegljivo: http://educacaofisicanamente.blogspot.si/2011/11/damas.html