

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 33 (2005/2006)

Številka 2

Strani 25-27

Branko Kaučič:

KAKO DELUJEJO ŠAHOVSKI PROGRAMI (PRVI DEL)

Ključne besede: računalništvo, miselne igre, programske rešitve, šah.

Elektronska verzija: <http://www.presek.si/33/1615-Kaucic.pdf>

© 2005 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

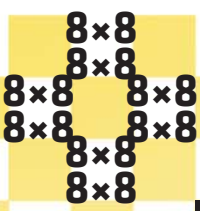
Kako delujejo šahovski programi (prvi del)

Branko Kaučič

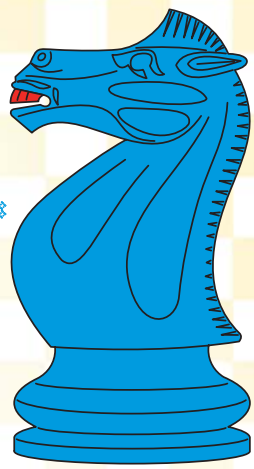
»Grega še eno partijo šaha?« Nekako tako je zvenelo iz ust mojega očeta, ko me je v osnovni šoli učil te kraljevske igre. Nekaj let intenzivnega igranja, polna glava miselnih vzorcev, uporaba računalnika in predvsem dejstvo, da očeta nikakor nisem uspel premagati, me je sčasoma privedlo do razmišljanja, kako napisati svoj šahovski program. Morda so podobno razmišljali tudi ljudje, ki so se podobnega podviga lotili že leta pred mano, za gotovo pa so se soočili s spoznanji, ki sledijo.

Šah je igra s »popolno informacijo«, ker sta oba igralca seznanjena s celotnim stanjem igre ves čas igranja. Že s pogledom na igralno ploščo je namreč razvidno, katere figure so še prisotne in kakšen je njihov položaj. Takšnih iger je še več (križci-krožci, go, backgammon), a igre, kot so popljanje ladjic in poker, ne sodijo v to kategorijo.

Pred vami je prvi prispevek iz serije prispevkov o računalniških programih za igranje šaha in strategij za podobne igre s popolno informacijo. Opisane bodo najpomembnejše tehnike, ki jih uporabljajo nekateri od najboljših šahovskih programov, vključno z legendarnim Deep Blue (več o tem v nadaljevanju). Večina tehnik, ki bodo omenjene, so namenjene skoraj vsem igram s popolno informacijo, čeprav se podrobnosti seveda razlikujejo od igre do igre. Poteze igralcev in ovrednotenje položaja so



Grega še eno partijo šaha?



6x6
alfa-beta
minmax
6x6

RAČUNALNIŠTVO

200.000.000 /s



namreč odvisni od pravil igre, ki jo igramo (oziroma poskušamo napisati računalniški program zanj).

■ Pomembnejši mejniki zgodovine šahovskih programov

Ideja stroja, ki obvlada šah, presenetljivo ne sovпада z začetki računalništva, temveč sega precej nazaj, v osemnajsto stoletje, povezana je z eno večjih prevar človeštva. Okoli leta 1776 je za zabavo cesarice Marije Terezije madžarski baron Wolfgang von Kempelen skonstruiral mehanski stroj. V njem se je skrival človek, večš mojstrovina šaha, ki je preko vzvodov premikal lutko Turka in s tem figure. Stroj je menjal lastnike, njegova skrivnost pa je dolgo ostala neodkrita.

Med prve stroje, ki so dejansko znali odigrati vsaj del partije šaha, štejemo mehanski stroj Ajedrecista Španca Torresa y Queveda. Stroj je znal odigrati končnico (zaključek igre) s kraljem in trdnjavo proti kralju. Leta 1912 je Torres y Quevedo predstavil prvo in leta 1920 izboljšano verzijo stroja. Stroja še danes delujeta v madridskem muzeju.

Prvi šahovski program je leta 1947 napisal matematik Alan Mathison Turing, čeprav zgolj na papirju. Moč računalnika je nadomestil kar sam in za izračun ene poteze porabil več kot pol ure. Leta 1949 je prav tako znamenit matematik Claude Shannon opisal, kako razviti šahovski program. Ugotovil je, da težavo predstavlja izredno veliko število izračunov položajev figur. Ločil je strategijo A, kjer pregledamo »vse« možne poteze (osnova algoritmu minmax), in strategijo B, kjer pregledamo le določene poteze. Na ta način tudi danes ločimo šahovske programe. S šahovnico velikosti 6×6 polj in brez lovcev so se na računalniku MANIAC ukvarjali tudi ameriški znanstveniki. Leta 1958 beležimo prvi pomembnejši dosežek treh znanstvenikov iz univerze Carnegie-Mellon (Newell, Shaw in Simon). Predlagali so postopek, imenovan algoritem alfa-beta, s katerim se je možno izogniti velike-

mu številu nepotrebnih izračunov. Podrobneje si ga bomo ogledali drugič. Istega leta je s pomočjo tega algoritma računalnik, s programom NSS, prvič premagal človeka.

Sledila je vrsta dosežkov, seznam nekaterih najde bralec v [1]. Med najodmevnejše med njimi sodi partija takratnega svetovnega šahovskega prvaka Garija Kasparova z računalnikom Deep Blue. Leta 1997 je računalnik zmagal z rezultatom 3,5:2,5, medtem ko je leto prej slavil Kasparov. Deep Blue je predstavljal paralelni računalnik s 30 vozlišči s skupaj 480 posebej namenskimi šahovskimi čipi. V tistem času je bil 259-ti najboljši računalnik na svetu. Šahovski program je bil napisan v programskem jeziku C in je zmožgal izračunati približno 200.000.000 položajev v sekundi, pri izboljševanju delovanja programa pa je sodelovalo kar nekaj velemojstrov.

■ Zgradba šahovskih programov

Uspešen šahovski program sestavljajo naslednje, med seboj tesno prepletene komponente:

- način, kako v pomnilniku računalnika predstavimo igralno ploščo (šahovnico), ki predstavlja stanje igre;
- pravila, ki določajo dovoljene poteze, da igramo brez goljufanja (in da preverjamo, da nasprotnik ne goljufa);
- tehniko, ki med vsemi dovoljenimi potezami izbere neko potezo (potezo raje izberemo po nekem ključu, kakor naključno);
- način primerjave različnih potez in položajev figur med seboj in
- uporabniški vmesnik.

V prispevkih bomo obravnavali vse zgoraj naštetu, razen uporabniškega vmesnika.

■ Predstavitev igralne plošče

V zgodnjem obdobju računalniških šahovskih programov je bila količina pomnilnika izredno omejena (programi so smeli uporabiti največ 8K pomnilnika) in najpreprostejše predstavitve šahovnice so bile najučinkovitejše. Le-ta je bila običajno kar polje velikosti 8×8 zlogov: prazno celico (brez figure) je



predstavlja vrednost 0, črnega kralja vrednost 1, črno kraljico vrednost 2. Z uvedbo 64 bitnih računalnikov so v takratni Sovjetski zvezi iznašli učinkovitejšo predstavitev, t. i. »bitne plošče«. Gre za 64 bitno besedo (1 bit za eno celico), ki vsebuje informacijo enega vidika stanja igre. Na primer, ena bitna plošča vsebuje »množico celic, ki jih zasedajo črni kmetje«, druga vsebuje »množico celic, kamor se lahko prestavi črna kraljica«, tretja vsebuje »množico celic, ki jih trenutno napadajo črni konji«. Bitne plošče so vsestranske in omogočajo hitrejše računanje položajev. Mnoge operacije, ki se med igro zelo pogosto ponavljajo, se namreč lahko implementirajo kot preproste logične operacije nad bitnimi ploščami.

■ Premikanje figur

Pravila igre določajo, katere poteze sme opraviti beli in katere črni igralec. V nekaterih igrah je že s pogledom na igralno ploščo možno določiti dovoljene poteze. Pri igri križci-krožci, na primer, vsaka prazna celica predstavlja dovoljeno potezo. Pri šahu, žal, ni tako preprosto. Vsaka figura ima svoja pravila premikanja in zajemanja drugih figur, prepovedano je pustiti kralja v šahu, poznamo tudi »en passant« poteze, promoviranje kmetov v višje figure, rokade, ki morajo ustrezati še dodatnim pogojem. Izkaže se, da je generiranje potez figur pravzaprav eden najtežjih vidikov (računsko zahtevnih in kompleksnih zaradi samih pravil) šahovskega programiranja. Pravila igre k sreči omogočajo, da nekatere operacije pripravimo vnaprej.

■ Iskalne tehnike

V šahovski program je treba zapisati tudi najbolj očitno. Računalnik namreč sam od sebe ne zna določiti, katera od dovoljenih potez je »dobra« in katera »slaba«. Velja preprosto razmišljanje: najlažje jih med seboj razlikujemo, če poznamo njihove posledice. Tvorimo zaporedje potez, na primer štiri za vsako igralno stran (skupaj torej osem) in pogledamo stanje igralne plošče. To je tudi osnovni princip algoritma minmax, ki predstavlja korenine vsakega šahovskega programa. Njegova učinkovitost je na žalost odvisna od povprečnega števila dovoljenih potez in števila potez, ki jih generiramo (imenovan tudi globina). Število možnih položajev tako raste izredno hitro in vložene ga je bilo precej truda

v algoritme, ki zmanjšajo število računanih položajev. Tak je na primer prej omenjeni algoritem alfa-beta.

Vzrok mnogih glavobolov šahovskih programerjev je tudi t. i. »problem horizonta«. Zamislite si, da vaš program išče osem potez vnaprej in ugotovi, da vam bo nasprotnik v osmi potezi vzal kraljico. Slabo stanje igre torej. Žal pa ne vidi, da bi to pravzaprav bila zvita poteza, s katero bi nasprotnik umaknil eno od svojih figur in vam v deveti potezi omogočil matiranje. Najboljše možno stanje igre torej. Namesto da bi izkoristil to priložnost, se bo program odločil raje za drugo potezo in tako »obvaroval« kraljico. Izkaže se, da je primerno število preverjenih potez težko določiti in uporaba vedno istega števila potez zelo verjetno vodi do slabe igre.

■ Ovrednotenje položaja figur

Program potrebuje tudi možnost ocenjevanja, ali določeni položaj figur vodi v zmago ali ne. Ovrednotenje je v veliki meri odvisno od pravil igre, najpomembnejše v šahu pa predstavlja število in tip figur na plošči. Že en kmet več lahko za dobrega igralca pomeni zagotovljeno zmago. Razviti uporabno funkcijo ovrednotenja položaja figur je težka naloga in predvsem to loči dobre šahovske programe od slabih.

Sedaj, ko v grobem poznamo vse komponente šahovske sestavljanke, se bomo v naslednjih prispevkih vsake od njih lotili nekoliko podrobneje.

■ Viri

- <http://www.geocities.com/SiliconValley/Lab/7378/comphis.htm>
- http://en.wikipedia.org/wiki/sahovski_racunalniski_programi
- <http://www.cs.biu.ac.il/~davoudo/tutorials.html>