Döntéstámogató rendszerek
2013/14 1.félév

Az IBM Deep Blue Sakkszámítógép

Készítette: Kohajda Dániel

2013.11.18.

Tartalomjegyzék

1. Bevezetés

A digitális számítógép feltalálása után, az 1950-es években komoly energiát fektettek a sakkszámítógépek fejlesztésébe. 70-es években a szakértők komolyan aggódni kezdtek a sakk gépek térhódítása miatt, és potenciális veszélynek érezték, hogy egy gép már legyőzheti az embert. 1978-ban David Levy, miután megverte a Chess 4.7-et, ami az akkori idő legerősebb sakk számítógépe volt, kijelentette, hogy az elkövetkező tíz évben nem lesz olyan gép, ami megverné őt.[1]

1989-ig kellett várni, hogy a kijelentése beigazolódjon, ugyanis az IBM Deep Thought sakkprogramja legyőzte egy bemutató mérkőzésen, majd 1996-ban az akkori sakk világbajnok, Garry Kasparov játszott az IBM Deep Blue sakkszámítógépével, amit ekkor sikeresen legyőzött. Ezt követően a Deep Blue-t jelentősen feljavították (nem hivatalos beceneve Deeper Blue lett**);** ez a gép játszott Kaszparovval 1997 májusában, amikor is nyert. Kasparov az IBM-et csalással vádolta és követelte a visszavágót, de az IBM ezt visszautasította. A számítógépnek a nyilvánosság előtt nem volt azóta komolyabb szereplése. Az IBM Yorktown-i épületében áll, ahol kutatási feladatokat végeznek vele, esetenként pedig különböző híresség elleni sakkjátszmákra használják.[2]

Garry Kasparov a Deep Blue ellen

2. Rendszer konfiguráció

Az 1997-es Deep Blue változat egy IBM RS/6000 SP alapú szuperszámítógép: masszívan párhuzamos megvalósítású, 30 db IBM RS/6000 SP processzort és összesen 480 speciális sakk-chipet tartalmazó gép. Minden processzor maximum 16 chipet tudott vezérelni, 8-8 chipet 2 mikro csatornán keresztül. Minden csomóponthoz tartozott 1Gb RAM és 4Gb háttértár. Mindegyik sakk-chip 2-2,5 millió (sakk)állást volt képes kiértékelni egy másodperc alatt, így az egész rendszer maximális teljesítménye kb. 1 milliárd állás/másodperc volt. A rendszer a gyakorlatban átlag 200 millió állás/másodperc sebességet ért el. Maga a sakkozó program C nyelven volt írva és AIX 4.2 operációs rendszer alatt futott. Az 1997-es játszma idején a keresőalgoritmus 40 lépéspár mélységig jutott el, míg a nem-kiterjesztett keresés 12 lépéspár mélységű volt.

A rendszer három rétegre osztható. Egy processzor volt kinevezve mesternek, míg a többi a feldolgozó. A mester feladata volt a játékfa felső szintjének a felderítése, majd a „levél" pozíciók szétosztása a feldolgozó processzorok között. A feldolgozók mélyebb szinteken folytatták a keresést, majd ezután továbbították az állapotot a sakk-chipeknek, amik sokszor nem végállapotok voltak, mivel a teljes játékfa felépítése túl sok idő lett volna. Emiatt nem a nyerő stratégia meghatározása volt a cél, hanem mindig a lehető legjobb lépés kiértékelése, majd meglépése. Ezt a feladatot a sakk-chipek látták el. [3][4]

3. A sakk-chip felépítése

A Deep Blue fejlesztése során sok korábbi sakkprogram kidolgozott ötleteit felhasználták fel, pl: a kereső algoritmust. Így próbálták a nagy és bonyolult rendszert szilárd alapokra helyezni.

A sakk-chip három részből állt, a **lépés generátorból**, a **kiértékelő függvényből** és a **keresés vezérlőből**. Ezeket alább részletesen bemutatom.

Lépés generátor

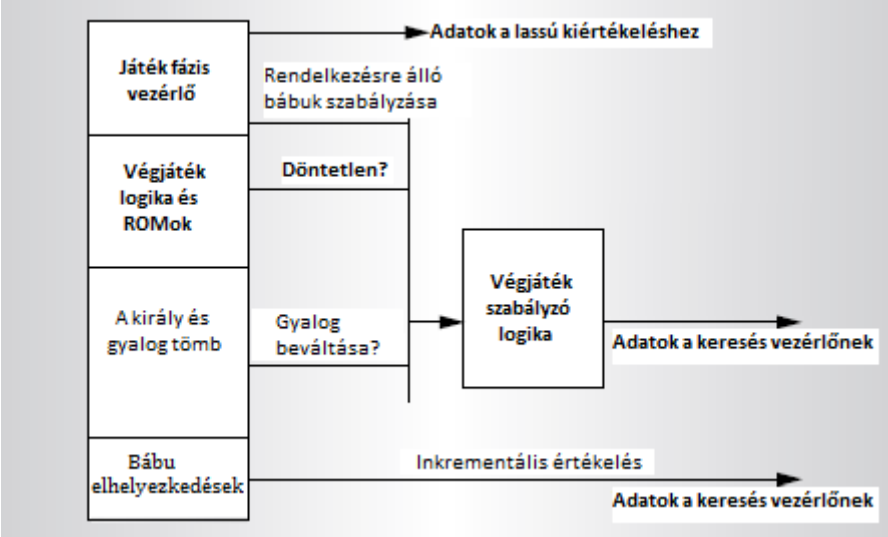
A Deep Blue **lépés generátorához** az IBM korábbi sakkszámítógépének, a Deep Thought-nak a lépés generátorát vették alapul, majd a chip-et kibővítették további funkciókkal, beépítettek generátor felügyelőt, támadó és védekező lépések bizonyos fajtáit is engedélyeztek generálni, ami javított a keresési algoritmus hatásfokán. A lépés generátor megvalósítása egy 8*8 soros kombinációs logikával történt, amely lényegében egy szilícium sakktábla. Minden cella négy fő részt tartalmaz: áldozat-kereső transzmittert, támadó-kereső transzmittert, vevőt és elosztott döntéshozót. Az áldozat keresésekor támadó jel kerül kisugárzásra. Ha az adott cellában nem tartózkodik bábu, a jel tovább terjed (bástya, királynő és futó áldozatainak keresése ezen az elven működik, gyalog és király esetén csak a szomszédos cellákba sugároz). Majd amikor a támadó jelek elérik a vevőt, megindul a feldolgozás és az áldozatokat rangsorolja, értékük szerint. A királynő a legmagasabb, a gyalogok a legalacsonyabb értékűek. Ezután a támadó keresés is lejátszódik, majd saját bábuit rangsorolja. Végül az elosztott döntéshozó dönt, a rangsor alapján, hogy milyen lépést kell megtennie.

A kiértékelő függvény

A kiértékelő függvény egy adott állásból kiindulva a játék várható hasznosságának becslését adja vissza. A becslés ötlete, amikor Shannon felvetette, nem számított újnak. A sakkozók (és természetesen más játékok szerelmesei) az évszázadok során kifejlesztettek olyan módszereket, amelyekkel mérlegelni tudták egy állás értékét, hiszen a keresés mennyiségének tekintetében az emberek még inkább korlátozottak, mint a számítógépes

programok. Nyilvánvaló, hogy egy játékprogram teljesítménye függ az alkalmazott kiértékelő függvény minőségétől. Egy pontatlan függvény a programot olyan állásokhoz vezethet, amelyek valójában vesztes pozíciók.

Milyen pontosan tudunk jó kiértékelő függvényeket tervezni? Először is a kiértékelő függvénynek a végállapotokat ugyanúgy kellene sorba rendeznie, mint az igazi hasznosságfüggvénynek, máskülönben az azt használó ágens még akkor is szuboptimális lépéseket választhatna, ha történetesen a játék végéig előre látna mindent. Másodsorban a kiértékelő függvény kiértékelése nem tarthat túl sokáig! Harmadsorban a nem-végállapotok tekintetében a kiértékelő függvénynek pontosan kell tükröznie a nyerés valódi esélyét. Előbbi három ok miatt a Deep Blue kiértékelő funkciójának végrehajtása kétféle lehet, gyors és lassú értékelés. Ez az általános technika, hogy elkerüljük a költséges kiértékeléseket, és hogy csak az adott játékmenethez tartozó hasznos kiértékelések fussanak végig. A gyors kiértékelés négy részből áll: a bábu elhelyezkedések táblázata, a király és gyalog tömb, a végjáték logika és ROM-ok, végül a játék fázis ellenőrzés. A király és gyalog tömb főleg a „gyalog beváltása" állapotot érzékeli. A végjáték logika és a ROM blokkja főleg a szokatlan végjáték feltételeket ismeri fel. A lassú kiértékelés egy időben csak egy oszlopot dolgoz fel; ide tartozik a király biztonságának az ellenőrzése, egyes bábuk csapdába esése, blokkolt gyalogok vizsgálata.



Keresés vezérlő

Az első cikk[1] a keresési algoritmusok témájában Claude Shannontól jelent meg, 1950-ben. Tökéletesen megjósolta, hogy két fő keresési stratégia irányába fognak elindulni, melyeket „A típusnak" és „B típusnak" nevezett el.

A „A típusú" programok nyers erőt, azaz kimerítő algoritmust használnak, mellyel az összes lehetséges lépést átnézik, például minimax algoritmussal. Shannon úgy vélte, hogy ez a stratégia nem hatékony, mivel a szükséges idő és adatterület lassúvá és nehézkessé tenné a programot.

1 Claude Shannon: *Programming a Computer for Playing Chess,* Philosophical Magazine, Bd.41, 1950.

Ahelyett, hogy elvesztegetné a processzor erőforrásait fölösleges vagy triviális lépések elemzésére, Shannon egy „B típusú” stratégiát is javasolt, mely az egyensúlyi keresést (quiescence search) alkalmazná, és szűkítené a keresési teret az eleve rossz találatok kiiktatásával.
Erre használják az **alfa-béta vágás** játékelméleti keresési algoritmust, amellyel csökkenthető a játékfában lévő kiértékelendő állások száma. Az algoritmus alapötlete azon nyugszik, hogy ha a játékfában az éppen vizsgált lépésünkre az ellenfélnek van egy olyan erős lépése ami miatt ezt a lépést úgyse választanánk (mivel a vizsgálat korábbi részéből már van jobb választásunk), akkor az erre a lépésre az ellenfél által adható további lépéseket nem szükséges megvizsgálni. Az algoritmusban ezen rész-játékfák fölösleges vizsgálatának kihagyását hívjuk alfa, illetve béta vágásnak.
A Deep Blue esetében a **keresés vezérlő** nem a hagyományos alfa-béta vágás kereső algoritmust használja. A minimális ablak keresés azon a feltételezésen alapul, hogy az összes részfa rosszabb, mint az eddig megtalált legjobb részfa, amíg ennek az ellenkezője be nem bizonyosodik. Miután a fa első ágait teljes szélességében megkereste, a keresés a többi ágban egy minimum alfa ablakkal [alfa, alfa+1] történik, ahol az alfa az eddig talált legnagyobb minimax értéket képviseli. Ha az ág által visszaadott érték kisebb vagy egyenlő mint az alfa, akkor a feltételezés helyes volt, a részfa gyengébb. Egyébként, ha a részfa nagyobb méretű, akkor a keresést újra kell kezdeni egy szélesebb minimum ablakkal, hogy a részfa pontos értékét kapjuk meg. Két ok miatt éri meg ennek a technikának az alkalmazása; először is könnyebb bebizonyítani, hogy egy részfa gyengébb, mint meghatározni annak pontos minimax értékét. Másodjára pedig, a legtöbb vizsgálat azt eredményezi, hogy a részfának felesleges a további kiértékelése. [5]

4. A rendszer „bevetése”

Bár a történelemben először fordult elő, hogy szabályos sakktornán számítógép emberi sakkvilágbajnokot győzött le, nincs még itt a világ vége. A Deep Blue brutális erőt képvisel ugyan (az IBM a hardvert akarja eladni, nem a szoftvert, ezért talán a kelleténél is több hangsúlyt helyeznek a nyers erejére és a klasszikus mesterséges intelligencia teljes hiányára), Kaszparov azonban még mindig jobb, vereségét főként tipikus emberi tényezők okozták. Már az első játékban rettenetesen elfáradt, és a szerencsével is végig hadilábon állt, ami egyre fokozódó pszichológiai nyomást jelentett. A második játszmában döntetlen állást adott fel, két nyerő állást sem tudott kihasználni, többször elnézett lépéseket, az utolsó játszmában pedig már teljesen maga alatt volt, gyerekes csapdába sétált bele. Szakértők szerint Kaszparov nem találta el igazán, mit kellene a számítógép ellen játszani; ebben persze az is közrejátszott, hogy a meccs előtt sosem látta még játszani a Deep Blue-t, így előzetesen csak "általában" a gépi játékstílusok ellen készülhetett fel. Ha megfelelően kiismerhette volna a számítógépet, vagy ha a játék tovább tartott volna, Kaszparov valószínűleg a Deep Blue fölé tudott volna kerekedni. Kritika érte Kaszparov megnyitási

taktikáját is: hogy Deep Blue-t idejekorán kizökkentse hatalmas megnyitási könyvtárából, Kaszparov második vonalbeli, meglepő lépései a játszmák elején nem kedveztek pozíciójának, ráadásul szokásos lehengerlő, támadó stílusa helyett passzív stratégiát választott, ami az eredmény ismeretében talán szintén hiba volt. Az idáig minden páros meccsén veretlen Kaszparov meg van győződve arról, hogy az IBM programozóit csupán az a cél vezette, hogy őt verjék meg.

5. Konklúzió

A Deep Blue 1997-es változata összesen 6 játékot játszott, mindet Kasparov ellen. Ezt a 6 játszmát a Deep Blue nyerte 3,5-2,5 arányban. Kasparov élő-pontszáma a játszmák idején 2815 volt, szemben a Deep Blue élő-pontszámával, amelyet körülbelül 2875-re becsültek. Azonban ezt a pontszámot nem lehet túl komolyan venni, mivel a minta mérete igen kicsi volt.

A Deep Blue programozói nem tudták volna megverni Kasparovot, azonban a Deep Blue-t felvértezték mindazzal a „tudással", ami elegendő volt egy élő ember sakk játszmában való legyőzéséhez. Mivel azonban a Deep Blue nem rendelkezik intelligenciával, tudását is csak az erős hardvernek és a szoftvernek köszönhette, amikkel másodpercenként több millió játékállás kiértékelésére volt képes és ezekből választotta ki a legjobbat. Egy-egy sakknagymester csak 50-100 ezer sémát tud hosszú távú memóriájában tárolni, illetve mozgósítani. Kijelenthető tehát, hogy a Deep Blue ember alkotta, ugyanakkor többet tud, mint az emberek, akik alkották. Igazából arról van szó, hogy hatalmas memóriájában sokkalta több információ van, mint az őt alkotó egyes emberek elméjében.

Kasparov valahogy úgy nyilatkozott arról a bizonyos lépésről, amikor a Deep Blue megverte, hogy „volt abban valami zseniális".

„Egy nagyon gyors tökfej." – mondta Garry Kasparov, 1996-ban, miután 4:2 arányban legyőzte az IBM Deep Blue nevű számítógépét.

6. Képjegyzék:

Előlap:
http://upload.wikimedia.org/wikipedia/commons/thumb/b/be/Deep_Blue.jpg/399px-Deep_Blue.jpg

http://hplusmagazine.com/wp-content/uploads/garry-kasparov-deep-blue-ibm.jpg

7. Irodalomjegyzék:

[1] Computer Chess History
http://www.worldchesschampions.com/software_chess_history.php

[2] Wikipédia
http://en.wikipedia.org/wiki/Deep_Blue_%28chess_computer%29

[3] Murray Campbell, A. Joseph Hoane Jr., Feng-hsiung Hsu: Deep Blue
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.2714&rep=rep1&type=pdf

[4] Feng-hsiung Hsu: IBM'S DEEP BLUE CHESS GRANDMASTER CHIPS
http://cseg.inaoep.mx/~jagonzalez/AI/Deepblue.pdf

[5] Minimal Window Search
http://aicat.inf.ed.ac.uk/entry.php?id=475

8. További források:

Sakk kifejezések szótára
http://www.eurochess.hu/aalap/modules.php?name=szotar&file=start

Fritz sakkprogramról
http://mialmanach.mit.bme.hu/erdekessegek/fritz_sakkprogramrol