

Šachová koncovka

kráľ, dáma proti kráľovi a veži

Zápočtový program pre zimný semester 2005 / 2006

Autor: Peter Černo, 1. ročník, MFF UK, informatika, skupina 37 (1-Y)

Názov programu: kqkr (king, queen VS king, rook)

Zdrojové súbory: kqkr.pas

Vývojové prostredie: Free Pascal 2.0.0

Platforma: nezávislá

Začiatok vývoja programu: 01.11.2005

Užívateľské prostredie

Spustenie programu (základné predvoľby):

kqkr.exe ... spustenie programu v základnom režime

kqkr.exe -cheat ... spustenie programu v rozšírenom režime

kqkr.exe -d ... debugovací režim určený na testovanie jednotlivých funkčných blokov programu (nie je zdokumentovaný)

Grafické rozhranie je najslabším článkom programu. Program pracuje v textovom režime, jednotlivé figúrky sú označené nasledovne:

K = king = kráľ

Q = queen = kráľovná

R = rook = veža

Ovládanie: Po spustení program overí, či existuje súbor strategy.dat. Pokiaľ neexistuje, spýta sa, či chcete uvedený súbor vygenerovať. Tento výpočet môže trvať na slabších strojoch niekoľko hodín. Pokiaľ tento súbor existuje, načíta sa do pamäti a následne sa spracuje.

V ďalšom kroku si vyberáte farbu: w (white) alebo b (black).

(biela farba = kráľ + dáma, čierna farba = kráľ + veža).

Po výbere program náhodne vyberie nejakú ťažkú pozíciu. Počas hry už iba zadávate ťahy a snažíte sa buď vyhrať (pokiaľ hráte za bieleho) alebo čo najdlhšie odolávať (keď hráte za čierneho).

Program ovládate zadávaním príkazov:

new - nová hra - znovu si vyberáte farbu (w / b)

exit - ukončenie programu

Zadávanie ťahov:

Povedzme, že chcete presunúť dámu (queen) z a3 na d6.

Máte dve možnosti:

a) zadať príkaz: qd6

b) zadať príkaz: a3d6

V prípade, že hráte za bieleho a dostanete sa do nedefinovanej pozície, program vypíše správu „game over“. Svojim ťahom ste totiž prišli o víťaznú stratégiu. (Taká pozícia môže nastať napríklad vtedy, keď nechtiac obetujete svoju dámu)

Rozšírený režim:

To isté ako základný režim, akurát máte niektoré príkazy navyše:

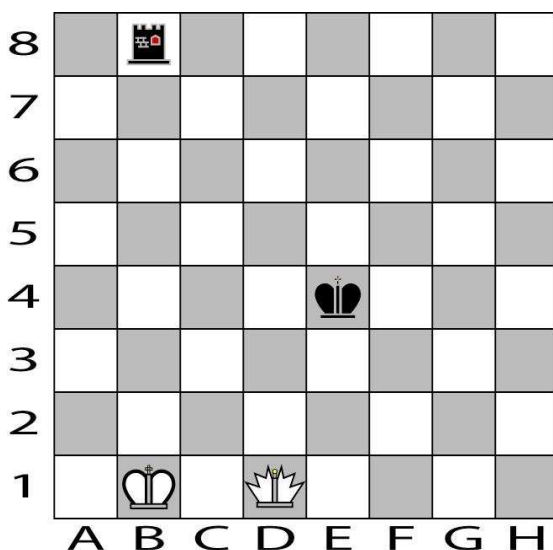
back - vrátenie posledne vykonaného ťahu

hint - počítač vám poradí optimálny ťah (najviac päť ťahov)

todo - zobrazenie počtu zostávajúcich ťahov do vašej výhry (biely)

alebo prehry (čierny)

Použité algoritmy



Obrázok 1: biely na ťahu

Na úvod: Bez ujmy na všeobecnosti predpokladajme, že biely hráč hrá s kráľom a dámou, čierny hráč hrá s kráľom a vežou.

Čiernu farbu budeme vždy označovať číslom 0, bielu číslom 1.

Pozíciou budeme rozumieť nejaké rozmiestnenie uvedených štyroch figúrok na šachovnici, pričom dáma a veža sa nemusia nutne na šachovnici vyskytovať.

Pozíciu nazývame pseudolegálna pozícia, pokiaľ:

- 1) sa na šachovnici vyskytuje biely aj čierny kráľ
- 2) žiadne dve figúrky sa nevyskytujú na tom istom políčku

Pozíciu nazývame legálna pozícia pre bieleho, pokiaľ:

- 1) je pseudolegálna
- 2) biely na ťahu nemôže vyhodiť čierneho kráľa

Pozíciu nazývame legálna pozícia pre čierneho, pokiaľ:

- 1) je pseudolegálna
- 2) čierny na ťahu nemôže vyhodiť bieleho kráľa

Na obrázku 1 vidíme pseudolegálnu pozíciu, ktorá je legálna pre bieleho, ale nie je legálna pre čierneho (čierny na ťahu totiž môže vyhodiť bieleho kráľa).

Pozíciu nazývame pre bieleho definovanú, pokiaľ:

- 1) je to legálna pozícia pre bieleho
- 2) biely na ťahu má v tejto pozícii víťaznú stratégiu

Všetky ostatné pozície sú pre bieleho nedefinované (obrázok 10, 11).

Podobne pozíciu nazývame pre čierneho definovanú, pokiaľ:

- 1) je to legálna pozícia pre čierneho
- 2) čierny je na ťahu a biely má víťaznú stratégiu (TJ po každom ťahu čierneho hráča sa biely hráč dostáva do víťaznej pozície)

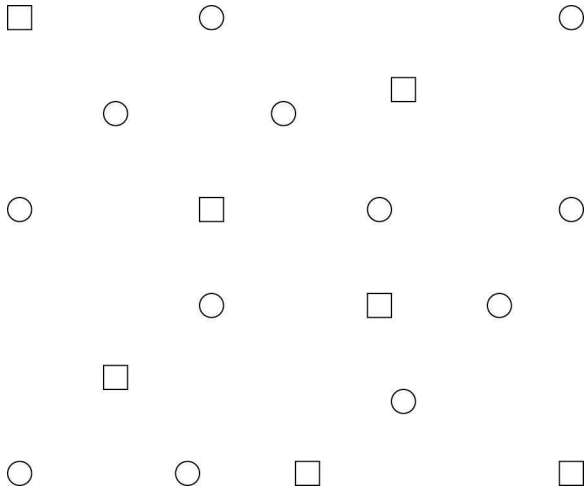
Všetky ostatné pozície sú pre čierneho nedefinované.

Napr. pozícia, v ktorej sa na šachovnici nevyskytuje biela dáma, je nedefinovaná pre čierneho aj pre bieleho.

Mojim cieľom bolo nájsť víťaznú stratégiu pre bieleho hráča.

Schopnosti programu: Program dokáže nájsť všetky optimálne ťahy pre bieleho (resp. čierneho) hráča v pozícii, ktorá je pre bieleho (resp. čierneho) definovaná.

Ako program funguje: Uvažujme množinu stavov, kde pod pojmom stav rozumieme usporiadanú dvojicu (pozícia, farba). Stav nám jednoznačne popisuje pozíciu a farbu hráča, ktorý je v tejto pozícii na ťahu. Všetkých stavov je dvakrát viac ako všetkých pozícií. Stav s farbou = 0 (TJ pozíciu s čiernym hráčom na ťahu) budeme označovať štvorčekom (= čierny stav), stav s farbou = 1 budeme označovať krúžkom (= biely stav).



Obrázok 2: stavový graf (hrany nie sú vykreslené)

Medzi jednotlivými stavmi môžeme nakresliť orientované hrany. Stav A je spojený orientovanou hranou so stavom B práve vtedy, keď existuje taký ťah, ktorým sa možno zo stavu A dostať do stavu B. Tento ťah musí samozrejme rešpektovať farbu, ktorá prislúcha stavu A. (TJ pokiaľ je stav A znázornený ako štvorec, pripúšťame iba ťahy čiernymi figúrkami). Lahko nahliadnuť, že orientovaná hrana spája vždy stavy rôznych farieb. (To sa dá interpretovať tak, že hráči sa striedajú v ťahoch). Dostávame teda niečo ako orientovaný graf na množine stavov. Tento graf budeme nazývať stavový graf (viď obrázok 2).

Ako nájsť optimálny ťah: Štandardným riešením by bolo použiť MINIMAX s alfa-beta odsekaním. Takéto riešenie je však úplne nepriateľné, pretože výpočet by vo väčšine prípadov neprebehol v reálnom čase. Existujú totiž pozície, kde by MINIMAX musel prehľadávať do hĺbky až 70 polťahov. (Polťahom rozumieme premiestnenie čiernej (resp. bielej) figúrky. Ťahom potom rozumieme dva po sebe idúce polťahy). Keď zoberieme do úvahy, že v priemere z každého stavu vychádza aspoň 20 orientovaných hrán, tak nám vyjde, že sme schopní prehľadávať maximálne tak do hĺbky 7 až 8 polťahov (vrátane odsekaní). Pokiaľ by sme sa uspokojili s „dobrým“ ťahom (TJ nie nutne optimálnym), mohli by sme použiť nasledujúcu fintu: našli by sme niečo ako oceňovaciu funkciu f , ktorá ku každému stavu X priradí nejaké reálne číslo. Čím je číslo $f(X)$ väčšie, tým je pozícia výhodnejšia pre bieleho a naopak. Kladné čísla $f(X)$ nám hovoria, že stav X je výhodný pre bieleho a nevýhodný pre čierneho. Napríklad pozícia, v ktorej biely dáva mat čiernemu je výhodná pre bieleho, a preto dostane vysoké ohodnotenie. Naopak pozícia, v ktorej má čierny vežu a kráľa, a biely má iba kráľa, je výhodná pre čierneho. Preto dostane takáto pozícia veľmi nízku (zápornú) hodnotu.

Výhoda tejto funkcie f je, že MINIMAX nemusí prehľadávať až k listom. Prehľadáva povedzme do vopred stanovenej hĺbky. (platí, že čím hlbšie prehľadáva, tým lepšie ťahy dokáže zahrať).

Nevýhodou funkcie f je, že danej pozícii priradzuje ocenenie na základe nejakej heuristiky. Môže sa stať (a stáva sa to často), že f priradí pozícii úplne nesprávne ocenenie. To sa potom odrazí na tom, že počítač zahrá zlý ťah.

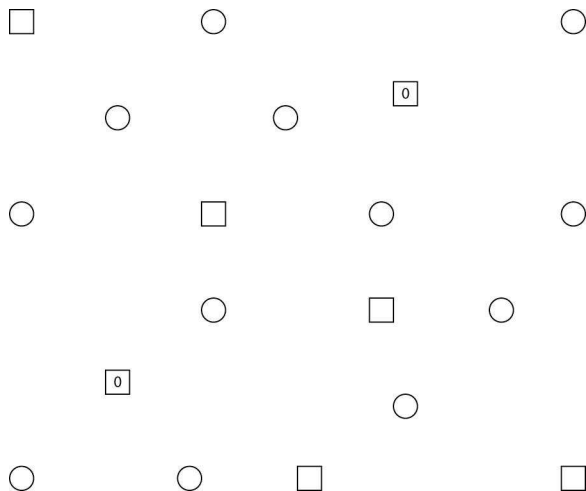
Existuje technológia tzv. hash tables, kde si v pomocnom poli uchováame, ktoré pozície MINIMAX ocenil. Keď potom MINIMAX narazí na nejakú pozíciu, ktorú kedysi už skúmal, iba sa pozrie do hash tabuľky a nemusí znovu prehľadávať to isté. Toto vylepšenie nám umožní prehľadávať povedzme do hĺbky až 15 polťahov, čo je značné zlepšenie. Bohužiaľ ani toto nestačí na nájdenie optimálneho ťahu.

Databáza STRATEGY: Budeme postupovať od listov smerom nahor. (Presne opačný postup, ako používa MINIMAX). Hľadáme víťaznú stratégiu pre bieleho, takže sa obmedzíme iba na stavy, v ktorých sa vyskytuje biela dáma. (Pozícia, v ktorej biely nemá dámu, totiž nie je pre bieleho víťazná. Môže byť buď remízová, alebo prehraná).

1) Najskôr označíme všetky stavy stavového grafu číslom 255.

(Stavy označené číslom 255 sú nedefinované)

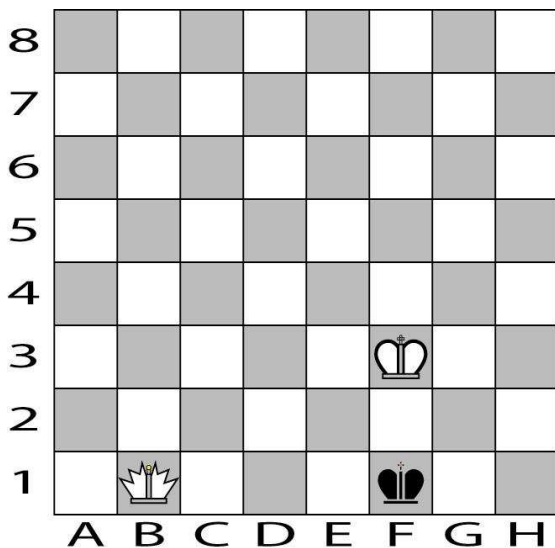
2) Všetky stavy, v ktorých má čierny hráč na ťahu mat, označíme nulou.



Obrázok 3

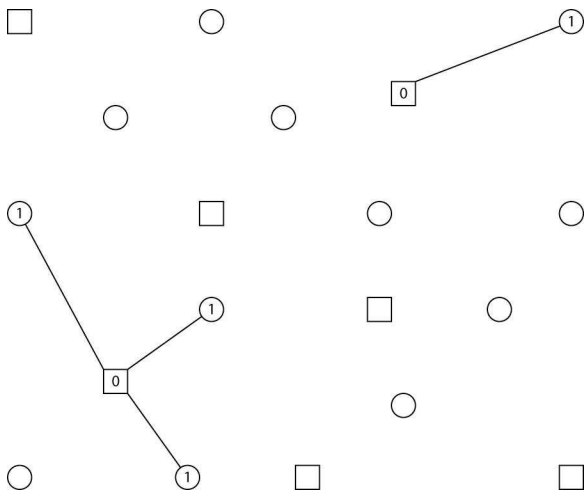
(Zrejme nulu môžu dostať iba štvorčky (čierne stavy) - TJ stavy, kde je čierny hráč na ťahu - viď obrázok 3)

Príklad pozície označenej číslom 0 je na obrázku 4:



Obrázok 4: Čierny na ťahu má mat

4) Ďalej označíme číslom 1 všetky biele stavy, z ktorých sa dá dostať do čierneho stavu s číslom 0. Sú to práve tie stavy, v ktorých biely na ťahu dokáže dať jedným ťahom (presnejšie polťahom) mat.



Obrázok 5

5) Ďalej sa budú striedať dva programy P0 a P1:

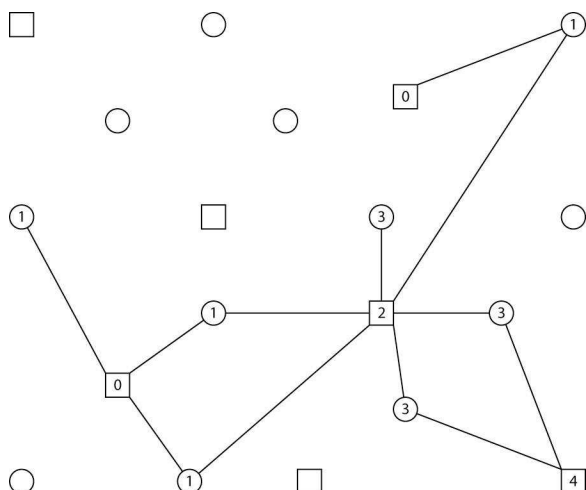
P0 - označí číslom 2 všetky čierne stavy X také, že každý ťah čierneho z X vedie do stavu s číslom 1.

P1 - označí číslom 3 všetky biele stavy X také, že existuje ťah bieleho z X do stavu označeného číslom 2.

Ďalej budú P0, P1 označovať stavy číslami 4 a 5, potom 6 a 7, atď. až nakoniec nezostane žiaden stav, ktorý treba ešte označiť.

Číslom 4 sú označené všetky čierne stavy X také, že každý ťah čierneho vedie do bieleho stavu Y s číslom 3 alebo 1 (pričom existuje aspoň jeden ťah taký, ktorý vedie do bieleho stavu s číslom 3).

Číslom 5 sú označené všetky biele stavy X také, že existuje ťah bieleho do čierneho stavu Y s číslom 4. Zároveň však neexistuje lepší ťah, TJ napr. ťah, ktorý by viedol do čierneho stavu Y s číslom 2 alebo 0.



Obrázok 6

Také jednoduché pozorovanie je, že čierne stavy dostanú párne čísla a biele stavy dostanú nepárne čísla.

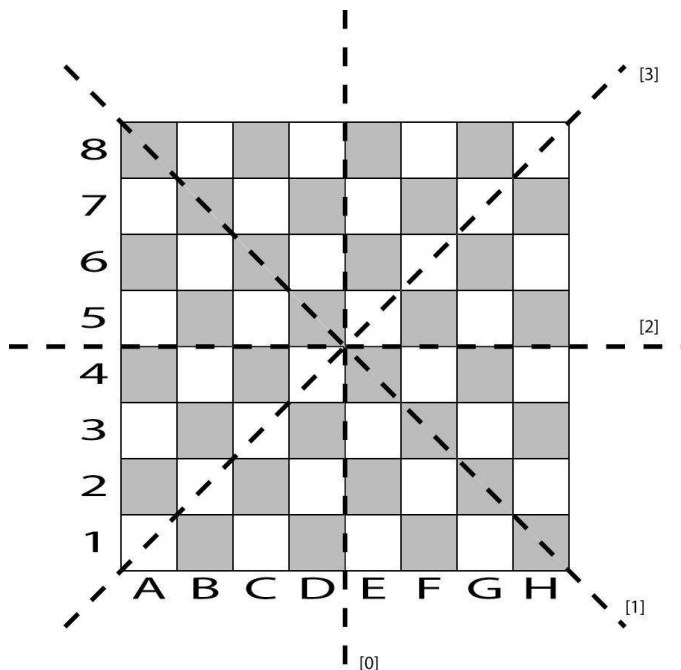
Poznámka: Toto je iba veľmi hrubý (a pomerne nepresný) náčrt použitého algoritmu. Je nutné ošetriť ešte zopár špeciálnych prípadov. Algoritmus použitý v mojom programe najskôr analyzuje pozície, kde sa vyskytuje biely kráľ, biela dáma a čierny kráľ; následne analyzuje pozície, kde sa vyskytujú všetky štyri figúrky (skrátí sa tým výpočtová doba).

Ako nájsť teraz optimálny ťah: Keď máme očíslované všetky definované stavy, nie je problém pre daný stav nájsť optimálny ťah.

Povedzme, že máme pozíciu X a hľadáme optimálny ťah pre bieleho. Predpokladáme, že pozícia X je pre bieleho definovaná. Stav $(X, 1)$ je potom priradené určité číslo rôzne od 255. Nech je to napríklad 15. (musí to byť nepárne číslo). Potom stačí nájsť (biely) ťah, ktorý nás dostane do pozície Y takej, že stav $(Y, 0)$ je priradené číslo 14. Existenciu tejto pozície máme zaručenú. Navyše sa nedokážeme dostať do pozície s číslom menším ako 14. To by muselo byť stav $(X, 1)$ priradené číslo nižšie ako 15. Keď zahráme zlý ťah, môžeme sa dostať buď do nedefinovaného stavu: $(Y, 0) = 255$, alebo do stavu s číslom $(Y, 0)$ väčším ako $(X, 1)$.

Ďalej nech máme pozíciu X a hľadáme optimálny ťah pre čierneho. Predpokladáme, že pozícia X je pre čierneho definovaná. Stav $(X, 0)$ je potom priradené určité číslo rôzne od 255. Nech je to napríklad 24. (musí to byť párne číslo). Potom stačí nájsť (čierny) ťah, ktorý nás dostane do pozície Y takej, že stav $(Y, 1)$ je priradené číslo 23. Opäť máme existenciu tejto pozície zaručenú. Pokiaľ by sme zahrali zlý ťah, mohli by sme spadnúť až napríklad na číslo 11. Nikdy sa nám však nepodariť dostať sa do pozície Y takej, že $(Y, 1)$ je väčšie ako $(X, 0)$. Dokonca keď hráme za čierneho, nikdy sa nám nepodariť dostať do nedefinovanej pozície pre bieleho, T. J. do $(Y, 1)$ rovné 255.

Vylúčenie symetrií: Na urýchlenie výpočtu vylúčime symetrické pozície. Ku každej pozícii X možno nájsť tzv. základnú pozíciu $f(X)$. Dve pozície X a Y sú symetrické práve vtedy, keď $f(X) = f(Y)$.



Obrázok 7: symetrie

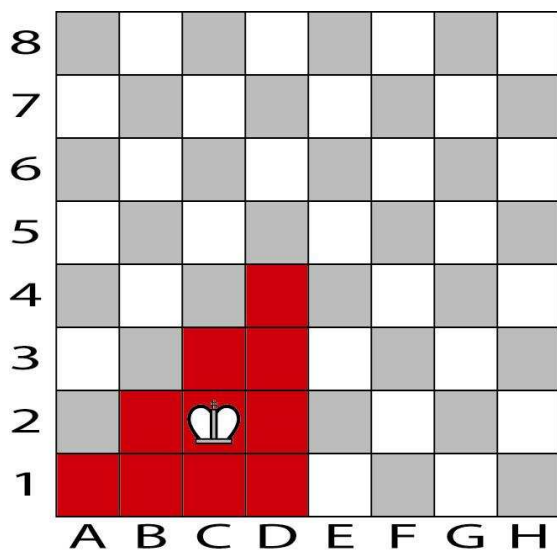
Keď máme dve symetrické pozície X a Y, potom možno jednu pozíciu previesť na druhú aplikovaním niektorých zo symetrií [0] až [3] (viď obrázok 7). Napríklad keď na pozíciu X aplikujeme postupne symetrie [0] a [1], dostaneme pozíciu Y, ktorá je pootočením pozície X o 90° proti smeru hodinových ručičiek.

Dá sa dokázať, že keď chceme previesť pozíciu X na symetrickú pozíciu Y, potom si vystačíme s nasledujúcimi 8 možnými postupmi:

0. nepouži žiadnu symetriu (v takom prípade je $X = Y$)
1. [0]
2. [1]
3. [2]
4. [3]
5. [0] [1]
6. [0] [2]
7. [0] [3]

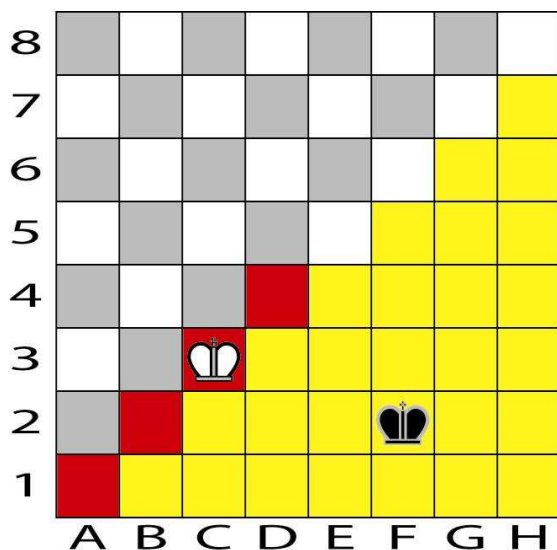
To znamená, že ku každej pozícii X existuje najviac 7 ďalších symetrických pozícií. (Dôkaz prenechávam čitateľovi).

Našou úlohou je teraz definovať pojem základná pozícia, (alebo tiež pojem: pozícia je v základnom stave)



Obrázok 8: základná pozícia

Keď máme nejakú pozíciu X , potom určite existuje postup, ktorým dostaneme bieleho kráľa do vyznačeného červeného trojuholníčka na obrázku 8. Podstatné je, že políčko v červenom trojuholníku je určené jednoznačne (TJ aplikovaním symetrií [0] až [3] nemôžeme bieleho kráľa presunúť z jedného červeného políčka do iného červeného políčka). Pokiaľ sa biely kráľ nachádza pod diagonálou (TJ na poliach B1, C1, D1, C2, D2, D3), potom hovoríme, že pozícia je v základnom stave. Keď sa biely kráľ nachádza na diagonále (TJ na poliach A1, B2, C3, D4), potom musíme rozlíšiť ešte niekoľko prípadov. Problém je totiž v tom, že keď v takomto prípade aplikujeme symetriu [3], poloha bieleho kráľa sa nezmení (TJ kráľ zostane v červenom trojuholníku), ale poloha ostatných figúrok sa zmeniť môže. Ako potom rozlíšiť základný stav? Odpoveď znie – na základe polohy ostatných figúrok.



Obrázok 9: základná pozícia

Predpokladajme, že sa biely kráľ nachádza na diagonále (A1 – D4, červené políčka), takže môžeme voľne aplikovať symetriu [3]. Potom pozícia je v základnom stave práve vtedy, keď:

1) čierny kráľ sa nachádza v žltom poli (viď obrázok 9)
 2) čierny kráľ sa nachádza na diagonále (A1 - H8) a biela dáma sa nachádza v žltom poli
 3a) čierny kráľ aj biela dáma sa nachádzajú na diagonále (A1 - H8), čierna veža sa nachádza v žltom poli
 3b) čierny kráľ sa nachádza na diagonále (A1 - H8), biela dáma sa nevyskytuje na šachovnici a čierna veža sa nachádza v žltom poli
 4) všetky figúrky sa nachádzajú na diagonále (A1 - H8)
 K bodu 4) treba dodať, že pod pojmom všetky figúrky rozumieme všetky figúrky, ktoré sa vyskytujú na šachovnici. Dáma a veža sa nemusia nutne na šachovnici nachádzať.
 Dá sa dokázať, že takto definovaný základný stav má požadované vlastnosti - TJ ku každej pozícii X existuje práve jedna symetrická pozícia $Y = f(X)$ v základnom stave.

Index pozície: Ku každej pozícii v základnom stave musíme vedieť jednoznačne priradiť nejaké číslo (z nejakého intervalu) - tzv. index pozície a naopak ku každému indexu (z nejakého intervalu) musíme vedieť jednoznačne priradiť príslušnú pozíciu v základnom stave.

Zaujímajú nás pritom iba tieto pozície:

- a) biely kráľ, biela dáma, čierny kráľ
- b) biely kráľ, biela dáma, čierny kráľ, čierna veža

Postup akým možno takúto bijekciu zostrojiť je pomerne dosť komplikovaný. Uvediem iba niektoré pozorovania:

Počet pozícií v základnom stave je presne:	2,131,072.
Z toho pseudolegálnych pozícií je presne :	1,937,880.
Počet pozícií legálnych pre bieleho je :	1,137,297.
Počet pozícií legálnych pre čierneho je :	1,375,962.
Počet pozícií, kde má biely mat je :	306.
Počet pozícií, kde má biely pat je :	0.
Počet pozícií, kde má čierny mat je :	1,420.
Počet pozícií, kde má čierny pat je :	109.

Poznámka: V definícii základnej pozície sme nepredpokladali, že figúrky sú umiestnené na navzájom rôznych políčkach - preto je počet pseudolegálnych pozícií menší ako počet všetkých pozícií. Zaujímavé je pozorovanie, že počet patových pozícií pre bieleho je nulový. To je zrejmé z toho, že uvažujeme iba pozície, kde biely disponuje dámou (a teda vždy môže vykonať nejaký ťah).

Otázka znie, k čomu sú nám pozície v základnom stave a na čo je dobrý index pozície? Základné pozície nám výrazne zredukujú veľkosť stavového grafu a index pozície nám uľahčí jeho implementáciu. Celý stavový graf potom môžeme zapísať ako pole:

```
var STRATEGY: array [0..1, 0..2131071] of Byte;
```

Kde prvý rozmer je pre farbu a druhý rozmer je pre pozíciu. Takéto pole potom takisto možno ľahko uložiť do súboru.

Štatistika databázy STRATEGY: Zaujímavé sú tiež niektoré pozorovania stavového grafu.

Vrcholy stavového grafu sú označené číslami v rozsahu 0 až 70, pričom nedefinované stavy majú číslo 255. Čierne stavy sú označené párnymi číslami, biele stavy sú označené nepárnymi číslami.

Počet všetkých vrcholov stavového grafu je: 4,262,144.

TJ presne dvakrát viac ako počet všetkých pozícií.

Počet stavov označených číslom 0 : 1,420.

Počet stavov označených číslom 1 : 11,591.

...

Počet stavov označených číslom 69 : 10.

Počet stavov označených číslom 70 : 59.

Počet bielych nedefinovaných stavov : 1,004,880.

Z toho legálnych pre bieleho je : 11,105.

Počet čiernych nedefinovaných stavov: 1,222,837.

Z toho legálnych pre čierneho je : 467,727.

Príklad stavu s číslom 69 je na obrázku 1.

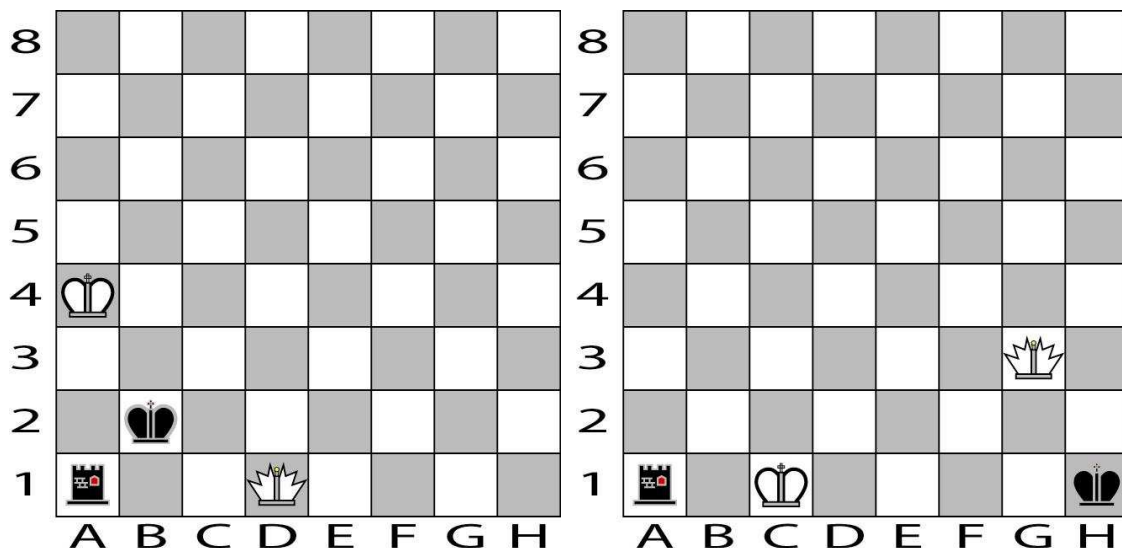
Všimnite si, že počet stavov označených číslom 0 je presne toľko, koľko je pozícií, kde má čierny mat.

Ďalej si všimnite, že legálnych nedefinovaných stavov pre bieleho je oveľa menej, než legálnych nedefinovaných stavov pre čierneho.

To je spôsobené tým, že každá legálna pozícia pre čierneho, v ktorej napríklad čierna veža ohrozuje bielu dámu, je nedefinovaná.

Vyplýva to z toho, že čierny na ťahu môže vyhodiť bielu dámu, čím biely stráca víťaznú stratégiu.

Príklad legálnej pozície pre bieleho, ktorá je zároveň nedefinovaná je na obrázku 10. Na obrázku 11 máme iný príklad - tu vidieť, že taká pozícia nemusí vyzerat' vôbec jednoducho. Nie je až také ťažké ukázať, že ani v tejto pozícii nemá biely víťaznú stratégiu. Stačí, keď čierna veža stále šachuje bieleho kráľa. V prípade, keď biely kráľ vyhodí čiernu vežu, nastáva pat.



Obrázok 10, 11: Príklad legálnej, nedefinovanej pozície pre bieleho.

Základné dátové štruktúry: Začneme reprezentáciou pozície figúrky na šachovnici. Poloha figúrky je jednoznačne určená číslom stĺpca (0..7) a číslom riadka (0..7).

Túto informáciu možno uložiť do jedného BYTU - dolné štyri bity udávajú číslo riadka, horné štyri bity udávajú číslo stĺpca. Napríklad v hexadecimálnom zápise predstavuje číslo 41 pozíciu E2.

Číslo FF je rezervované pre nedefinovanú pozíciu - indikuje, že daná figúrka je vyhodená.

Typ TState:

```
type TState = record
    {white king,  black king,
     white queen, black rook}
    WK, BK, WQ, BR: Byte;
end;
```

TState nám jednoznačne popisuje pozíciu, T.J rozmiestnenie jednotlivých figúrok na šachovnici. (Nenechajte sa zmiast' nevhodným názvom TState)

Typ TList:

```
type PCell = ^TCell;
TCell = record
    mState: TState;
    pNext: PCell;
end;
TList = object
    pHead: PCell;
    Count: LongInt;
    procedure Init;
    procedure Free;
    function IsEmpty: Boolean;
    procedure AddState(var State: TState);
    procedure AddList (var List: TList);
    procedure Pop;
end;
```

TList predstavuje spájaný zoznam pozícií (TState).

Niektoré pomocné konštanty a premenné:

```
const
    set01      : set of Byte = [0, 1];
    set23      : set of Byte = [2, 3];
    set23X     : set of Byte = [2, 3, 255];
    set01234   : set of Byte = [0, 1, 2, 3, 4];
    set0123X   : set of Byte = [0, 1, 2, 3, 255];
    Range0     = 0;
    Range1     = 24576;
    Range2     = 31744;
    Range3     = 32640;
    Range4     = 32896;
    Range5     = 1605760;
    Range6     = 2064512;
    Range7     = 2121856;
    Range8     = 2129024;
    Range9     = 2131072;
    RangeKQKX  = Range4;
    RangeKQKR  = Range9;
```

Konštanty Range0 .. Range9 sa používajú pri prevode pozície na index a naopak. Množiny setXXXX sa používajú pri zistení základnej pozície.

```
var BasicTable: array[Byte] of Byte;
    OffsetA: array[Byte] of Byte;
```

```

OffsetB:      array[Byte] of Byte;
OffsetC:      array[Byte] of Byte;
InvOffsetA:   array[Byte] of Byte;
InvOffsetB:   array[Byte] of Byte;
InvOffsetC:   array[Byte] of Byte;

```

Pomocné polia na prevod pozície na index a naopak + na zistenie základnej pozície.

```

var   STRATEGY:      array[0..1, 0..RangeKQKR - 1] of Byte;
      STRATEGYCheck: Boolean;

```

Pole STRATEGY predstavuje náš stavový graf, každému stavu priraduje nejaké číslo.

STRATEGYCheck indikuje, či je pole STRATEGY naplnené datami.

Základné procedúry a funkcie:

```

function Row(X: Byte): Byte;
Určí číslo riadka (0..7) pre pozíciu X.

```

```

function Col(X: Byte): Byte;
Určí číslo stĺpca (0..7) pre pozíciu X.

```

```

function Add(r, c: Byte): Byte;
Na základe čísla riadka r a čísla stĺpca c určí pozíciu (BYTE).

```

```

procedure InitVariables;
Inicializuje niektoré pomocné premenné.

```

```

function Convert(State: TState): LongInt;
Prevedie pozíciu v základnom stave State na index.

```

```

function InvConvert(Index: LongInt): TState;
Prevedie index na pozíciu v základnom stave.

```

```

procedure Sym(var X: Byte; Index: Byte);
Na pozíciu figúrky X aplikuje symetriu [Index].

```

```

procedure SymState(var State: TState; Index: Byte);
Na pozíciu State aplikuje symetriu [Index].

```

```

function IsBasicState(var State: TState): Boolean;
Zistí, či pozícia State je v základnom stave.

```

```

procedure BasicState(var State: TState; var Epsilon: Byte);
Prevedie pozíciu State na pozíciu v základnom stave.
Epsilon určuje, ktorý z ôsmych postupov pri prevode procedúra
BasicState aplikovala.

```

```

procedure OriginalState(var State: TState; Epsilon: Byte);
Prevedie pozíciu v základnom stave State na pôvodnú pozíciu na základe
čísła postupu - Epsilon.

```

```

function IsAttacked(X: Byte; State: TState; Colour: Byte): Boolean;
Zistí, či farba Colour atakuje políčko X v pozícii State.

```

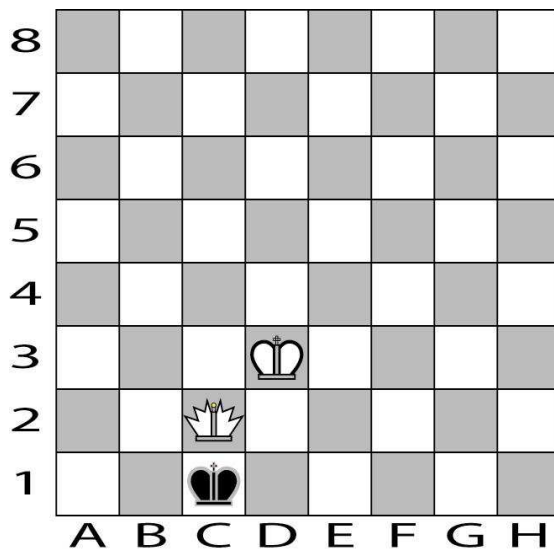
function IsPseudoLegal(State: TState): Boolean;
Zistí, či je pozícia State pseudolegálna.

function IsLegal(State: TState; Colour: Byte): Boolean;
Zistí, či je pozícia State legálna pre farbu Colour.

procedure Generator(**var** List: TList; State: TState; Colour: Byte);
Z danej pozície State vygeneruje všetky možné ťahy (farba hráča na ťahu je Colour).
Pozície, do ktorých sa možno dostať uloží do zoznamu List. Uloží tam iba tie pozície, ktoré sú legálne pre hráča s farbou 1-Colour.
Funkčnosť tejto procedúry možno otestovať v debugovacom režíme.

procedure PseudoGenerator(**var** List: TList; State: TState; Colour: Byte);
Z danej pozície State vygeneruje všetky možné pozície X také, že z pozície X sa môže hráč s farbou Colour dostať do pôvodnej pozície State. Do zoznamu List potom uloží iba tie pozície X, ktoré sú navyše legálne pre hráča s farbou Colour.
Funkčnosť tejto procedúry možno otestovať v debugovacom režíme.

Uvažujme napríklad pozíciu ako je na obrázku 12:

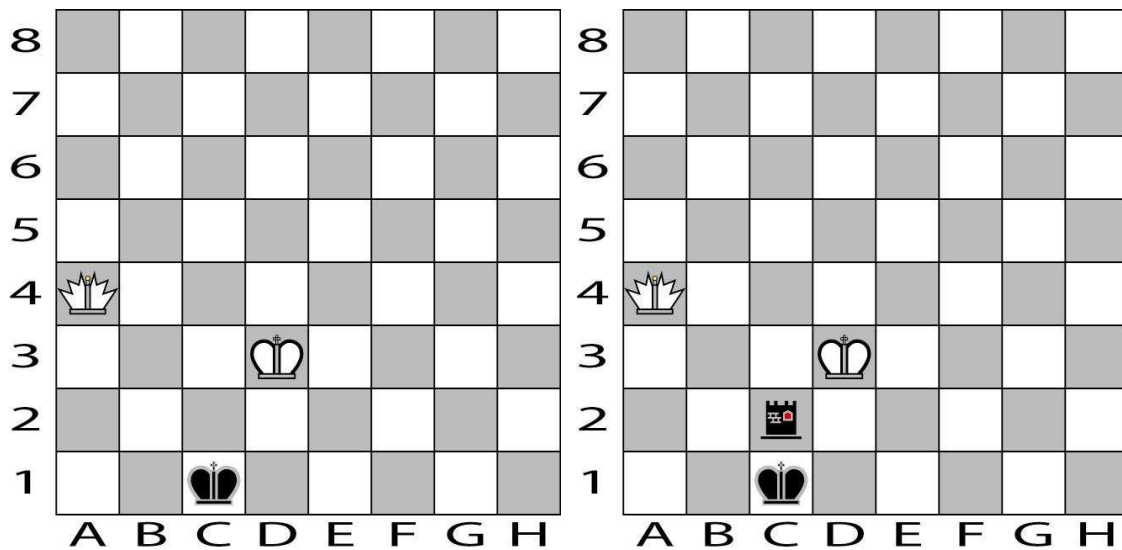


Obrázok 12

Hľadáme pseudo-pozície pre bieleho hráča.

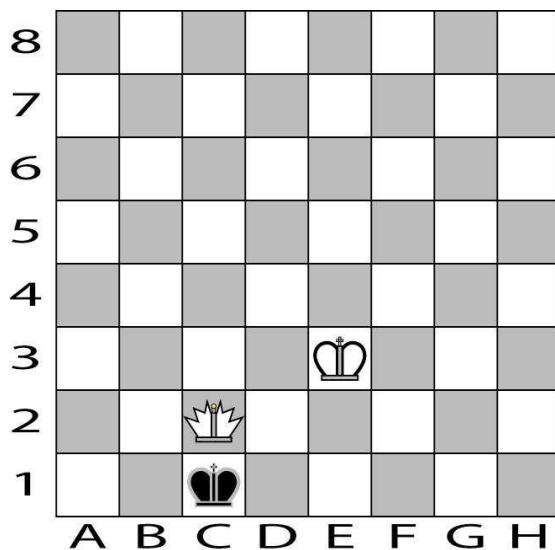
(Pre čierneho hráča zrejme žiadne nie sú).

Do tejto pozície (ako je na obrázku 12) sa možno dostať napríklad z pozície na obrázku 13, ale aj z pozície na obrázku 14:



Obrázok 13, 14

Avšak nemožno sa tam dostať z pozície ako je na obrázku 15.



Obrázok 15

Pretože táto pozícia nie je legálna pre bieleho hráča.

Všimnite si, že pozícia na obrázku 12 by v databáze STRATEGY dostala číslo 0 (pre čierneho hráča na ťahu), pozície na obrázkoch 13 a 14 by dostali číslo 1 (pre bieleho hráča na ťahu).

function IsLegalMove(FromState, ToState: TState): Boolean;
Zistí, či sa možno (jedným legálnym ťahom) dostať z pozície FromState do pozície ToState.

function MoveToStr(FromState, ToState: TState): string[10];
Prevedie ťah z pozície FromState do pozície ToState na reťazec.

function StrToMove(var ToState: TState; FromState: TState; Colour: Byte; var S: string[50]): Boolean;

Prevedie reťazec S na ťah z pozície FromState. Výstup je výsledná pozícia ToState.

procedure OptimalMove(**var** List: TList; **var** State: TState; Colour: Byte);
Nájde všetky optimálne ťahy v pozícii State pre farbu hráča Colour a uloží ich do zoznamu List.

function SaveSTRATEGY: Boolean;
Uloží pole STRATEGY do súboru strategy.dat.

function LoadSTRATEGY: Boolean;
Načíta pole STRATEGY zo súboru strategy.dat

procedure STRATEGYGenerator;
Vygeneruje pole STRATEGY.

procedure PrintState(State: TState; X0, Y0: Byte; DEBUG: Byte);
Zobrazí pozíciu State na obrazovku s počiatočnými súradnicami X0, Y0. Príznak DEBUG je určený na testovanie – používa sa v debugovacom režime.

procedure Tester;
Spustí užívateľské rozhranie debugovacieho režimu.

procedure Game;
Spustí užívateľské rozhranie hracieho režimu.

Záverečné poznámky: Na tejto koncovke ma najviac zaujalo to, že napriek tomu, že má biely hráč víťaznú stratégiu, nie je ľahké čierneho hráča poraziť. Je to dokonca tak náročné, že si s touto koncovkou neporadili ani mnohí šachoví veľmajstri.
Pri písaní programu som vychádzal z materiálov pripojených k tejto dokumentácii.

Databáze koncovek

V této pasáži nepůjde o programování šachu v tradičním slova smyslu, ani o špičku různých příkladů koncovek, jak byste se snad mohli na základě předchozích informací o šachových databázích domnívat. V případě "databáze koncovek" jde o úplnou analýzu koncovek, která je proveditelná jen díky obrovskému výpočetnímu výkonu počítačů. Neprovádí se propočty variant ani oceňování pozic. Program se dívá do - jak uvidíme opravdu mohutného - seznamu a hraje perfektní tahy. Na datovém nosiči jsou totiž uloženy *veškeré pozice*, jaké mohou v určité koncovce nastat.

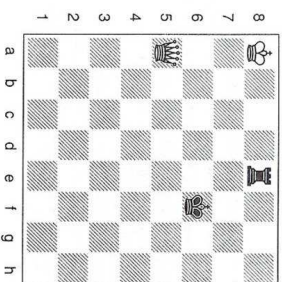
Historie databází koncovek začala v roce 1983 v jedné počítačové laboratoři na východě Spojených států.

Sázka platí

Koncovka král a dáma proti králi a věži vyhřává strana s dámou - to je v šachové teorii známo už po staletí. Ale postup k vítězství není tak úplně triviální. Specialista na koncovky André Chéron potřeboval ve své "Účebnici a příručce koncovek" plných sedm stran na vysvětlení správné strategie. Přesto si každý turnajový hráč bezesvého troufne na dobytí protivnickovy věže dámou. Pravidla šachu předpokládají, že k tomu musí dojít během 50 tahů - hráč, který dokáže, že po dobu 50 tahů nebyl brán jediný kámen a nebylo taženo pěšcem, může ihned reklamovat remízu. Většinou je ale vše skončeno po 10 až 20 tazích.

Proto byli účastníci velmistrovského turnaje, který se konal počátkem 80. let v USA, překvapení nabídkou počítačového experta, že dá 100 dolarů tomu, kdo nastoupí s dámou v této koncovce proti jeho novému programu. Ken Thompson, konstruktér Belle a autor Unixu, vyvinul v Bell Laboratories v New Jersey program, který měl údajně perfektně ovládat koncovku KD-KV. Většina z přítomných velmistrů považovala tuto výzvu za žert a trojnásobný šampion USA Walter Browne sázku ihned přijal. Podmínky byly pro něj velmi výhodné: na 50 tahů až do vítězství věže nebo matu dostal Browne dvě a půl hodiny, kromě toho se mohl na tuto koncovku speciálně připravovat a navíc libovolně dlouho před začátkem hry analyzovat připravovanou pozici. Naproti tomu měl Thompsonův počítač na každý tah jen jednu sekundu a při každém tahu protivníka uváděl, kolik tahů potřebuje při optimální hře blýst k vítězství (v následující notaci v závorkách). Historické střetnutí mělo následující průběh.

velmistř Walter Browne - Thompsonova databáze



1. Kb7 (31) Ve7+ 2. Kc6 (30) Ve6+ 3. Kd7 (29) Ve7+ 4. Kd8 (28) Ve4 5. Dc5 (27) Ve5 6. Dd4 (26) Kf5 7. Kd7 (25) Ve4

8. Dd3 (24) Kf4 9. Kd6 (23) Ve3 10. Dd4+ (22) Ve4 11. Df2+ (21) Kg4 12. Kd5 (20) Ve8 13. Df6 (19) Ve3 14. Kd4 (18)

Vf3 15. Dg6+ (17) Kf4 16. Dg2 (16) Va3. Až doposud hrál velmistř optimálně. Ale nyní udělal chybu: 17. Dc6? (20) Va1 18. Dc7+? (21) Kf5 19. Dc2+ (20) Ke6. Divák usuzuje, že Browne chce dobýt věž vidličkou. Na to Thompson: "It can't be done". 20. Dd2 (19) Va7

21. Db4 (21) Ve7 22. Ke4 (20) Kf6+ 23. Kf4 (19) Ke6 24. Dd4 (18) Vf7+ 25. Ke4 (17) Vf6 26. Dd5+ (16) Ke7 27. Ke5 (15) Vh6 28. Db7+ (14) Kd8. Velmistř dělá opět pokroky, ale Thompson zůstává plný důvěry: většina lidí selhala právě ve chvíli, kdy k vítězství zbývalo 17 až 14 tahů. 29. Df7 (14) Ve6 30. Kd5 (15) Vb6. Browne: "Nějak se mu daří držet pozici. To v žádné knize není." 31. Ke5 (14) Va6 32. Dc4 (17) Vf6 33. Db4 (17) Ke7

34. Kd5 (17) Kf7. Thompson: "Už je to pryč. Počítač říká, že už nemůžete vyhrát". Browne: "Počítač se může mýlit!" 35. Ke5 Ve6+ 36. Kf5 Vd6 37. Dc4+ Ke7 38. Ke5 Vh6 39. Dc7+ Kf8 40. Kf5 Ke8 41. De1 Vd6. Nyní Browne pozná, že už to nedokáže, ale zahrál ještě pár tahů. 42. Dc8+ Ke7 43. Dc7+ Vd7 44. Dc5+ Kd8 45. Kc6 Vb7.

Velmistr přerušil partii a chtěl vidět, jak by hrál počítač blýmsi. Během sekundy tu byla analýza s optimální hrou pro obě strany:

1. Ka7 Ve7+ 2. Kb8 Ve8+ 3. Kc7 Ve7+ 4. Kd8 Ve4 5. Da8 Ve3 6. Kc7 Ke5 7. Da5+ Ke4 8. Kd6 Kf4 9. Dh5 Vd3+ 10. Kc5 Va3 11. Dh2+ Kg3 12. Dd2+ Kf5 13. Dc2+

8. Počítačový šach pro profesionály

275

Kg4 14. Kd4 Vf3 15. Dg2+ Vg3 16. Dc2+ Kg5 17. Dc5+ Kh4 18. De1 Kf3 19. Dh1+ Kg4 20. Ke4 Kg5 21. Dh2 Kg4 22. Dh6 Vg2 23. Dg6+ Kh3 24. Dh5+ Kg3 25. Ke3 Vg1 26. Dc5+ Kg2 27. Ke2 Vh1 28. Dc4+ Kg1 29. Dg4+ Kh2 30. Kf2 Va1 31. Dg2 mat.¹³

Vyčerpávající analýza

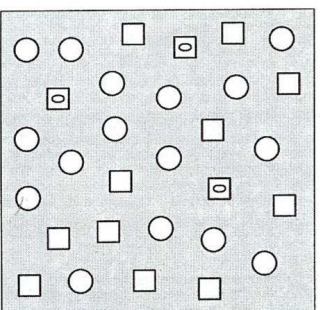
Čtenář si jistě všiml, že počítač ve výše uvedeném případě předpovídal mat do 31 tahů. Program přesně věděl hned ve výchozí pozici a také během celé partie, kolik tahů je nutných při optimální hře. Jak mohl Thompsonův program tak perfektně ověřit koncovku?

Tento počítačový vědec nastoupil se svými databázemi koncovek cestu, která nemá s obvyklými technikami programování šachu nic společného. Jeho počítač totiž před provedením každého tahu vyčerpávajícím způsobem analyzoval kompletní koncovku a výsledky uložil na pevném disku.

Analýza byla provedena za pomoci velkého počítače. Ten nejprve vygeneroval všechny legální pozice, které zadane čtyři kameny umožňují. Zahnuje to i pozice se třemi kameny, tedy takové, v nichž chybí dáma nebo věž. Celkem to je 1900000 pozic, z nichž ještě nebyly vytvořeny nepřipustné konfigurace (dva kameny na stejném poli, králové vedle sebe). Pro ocenění pozice použil Thompson speciální "retro-

Obrázek 8.12

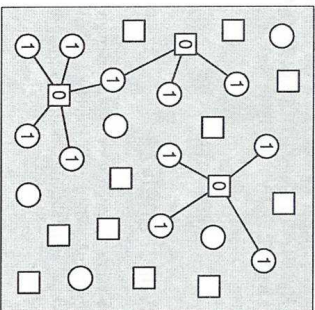
Pozice, v nichž černý dostává mat, jsou označeny nulou. Pozice s čermým na tahu jsou znázorněny jako čtverec, pozice s blým na tahu jako kruh.



¹³ Walter Browne se nemohl se svojí porážkou smířit a žádal odvetu. Po deseti letech připravě opět nastoupil proti počítači. Tentokrát to dokázal, ovšem velmi těsně. Přesně v 50. tahu se velmistřovi podařilo dobýt protivníkovu věž.

Obrázek 8.13

Program P2
označuje
všechny
pozice, v nichž
může dát bílý
jedním tahem
mat, jedničkou.

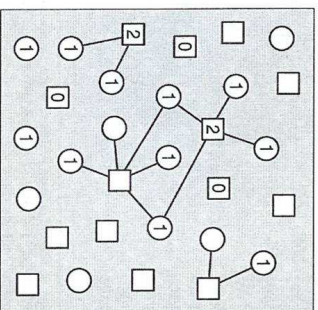


analytický" postup, v němž jistým způsobem zpětně - vycházejíc z konečné pozice - prozkoumal a ocenil všechny pozice.

Provádějí to čtyři programy - P1 až P4. Program P1 nejprve prozkoumá všechny pozice se třemi kameny (král, dáma a král) a označí mlou ty z nich, v nichž černý dosel mat (obrázek 8.12). Potom program P2 vytvoří všechny legální pozice s bílým na tahu, které vedou jedním tahem k matu. Ty pak v souboru pozic označil jedničkou. Jsou to tedy pozice, v nichž bílý dává jedním tahem mat (obrázek 8.13). P3 pak vzal všechny tahy předcházející pozici označené jedničkou a vytvořil z nich pomocný soubor J1. Tento pomocný soubor tedy obsahoval pozice, ve kterých mohl černý jed-

Obrázek 8.14

Programy P3
a P4 zjišťují
pozice
s černým na
tahu, v nichž
veškeré tahy
vedou k pozici
označené jed-
ničkou. Takové
pozice jsou pak
indikovány
dvíčkou.

**8. Počítačový šach pro profesionály**

277

ním tahem přivodit pozici označenou jedničkou (černý měl samozřejmě většinou mnoho jiných tahů, které k takovým pozicím nevedly).

Potom vytvořil program P4 pro každou pozici v J1 všechny následovníky a prověřil stav těchto pozic v hlavním souboru. Pokud byly všechny označeny jedničkou, obdržela pozice J1 dvojku. Byly to pozice, v nichž nemohl černý žádnou obranou zabránit tomu, aby mu bílý nedal následujícím tahem mat (viz obrázek 8.14).

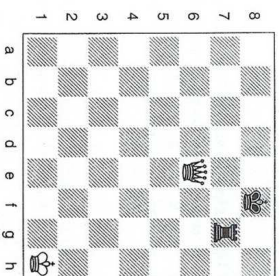
Potom byly programy P2, P3 a P4 spuštěny znovu, aby označily pozice se třemi a čtyřmi pěti tahy do matu nebo ztráty věže. To se opakovalo tolikrát, dokud už nebyly nalezeny nové pozice (se třemi kameny). Přitom se přirozeně dávalo pozor na to, aby žádná pozice nebyla označena podruhé: když nějaká pozice vede k matu třetím tahem, pak nemá význam to, že je možný mat i sedmým tahem.

Po určité době byly oceněny všechny pozice s králem a dámou proti králi. Nyní začaly programy P1 až P4 se zkoumáním pozic se čtyřmi kameny. P1 označil všechny maty, P2 zanesl jedničku za každou pozici, v níž mohl bílý přivodit jedním tahem mat. P3 a P4 zjišťovaly všechny pozice, v nichž musel černý připustit mat jedním tahem nebo vyhranou pozici se třemi kameny. P2, P3 a P4 běžely tak dlouho, dokud nebyly označeny žádné nové pozice.

Půl dne trvalo, než velký počítač prozkoumal všech 1,9 miliónu pozic koncovky král a dáma proti králi a věži. Hodně přes 1,7 miliónu z nich obdrželo číslo mezi 0 a 61, což znamená, že maximální délka vyhrané partie v této koncovce obnáší 31 tahů. Existují ostatně jen dvě pozice, v nichž je skutečně nutných tolik tahů (jedna z nich je výše uvedená východní pozice partie Brownea proti Thompsonové databázi). Vzhledem k tomu, že bylo označeno více než 90% všech pozic, může být tato koncovka v zásadě označena jako "vítězná pro bílého". Výsledek Thompsonova zkoumání tedy zněl: V koncovce KD-KV může strana s dámou zpravidla dosáhnout výhry (což se už vědělo), a sice v 61 nebo méně pultazích (to zatím známo nebylo).

Co ale se zbývajícími pozicemi, které nejsou nijak označeny? Většinou jsou to takzvané "patologické" pozice, v nichž může černý krátkodobě vyměnit věž za bílou dámu. Například v pozici bKb1, Dal, eKb8. Vhl nemůže bílý samozřejmě vyhrát, a když čeká místo na b1 stojí na d1, e1 nebo f1, pak dokonce vyhává černý. Existují také dlouhodobě remízové pozice, v nichž se objevují patové motivy. Zde je jeden příklad, který byl znám už před dvěma sty lety:

Ponziani, 1782



Černý je na tahu a udělí remízu

Věž dává na h7, g7 a f7 šach. Bílý král samozřejmě nemůže kvůli Ve7 vsioutit do linie e, a když vezme věž na poli h7, má černý král pat. Jestliže naproti tomu přejde bílý král na f6, přijde Vg6+ remíza!

Takové pozice jsou v databázi označeny jako remíza, nebo jsou prostě vypuštěny. Pokud počítač naráží na remízovou pozici, musí jako obhranec dávat pozor na to, aby hrál jen tahy, které rovněž vedou na remízové pozice. Jako útočnick bude hrát dost nepřálovitě. Protože žádný tah nevede k vítězství, vybere se náhodně, takže za jistých okolností hraje bílý ve výše uvedené pozici takto: 1...Vh7+ 2.Kg1 Vg7+ 3.Kh2 Vh7+ 4.Dh3. Proč si nechávat dámu - remíza nakonec zůstane remízou!

Hrát jako bůh z New Jersey

Bohužel je nanejdůležitější obtížné odvodit z postupu počítače všeobecnou strategii, kterou by mohli použít lidé. Jestliže si přehrajete výše citovanou partii dány proti věži, uvidíte, že počítač neudrží nutně věž v blízkosti krále, jak to vyžaduje teorie. Naopak se zde objevuje situace, kdy věž vytváří bariéru v šesti řadě, kterou musí útočník překonat - a o tom se v žádné teoretické knize nehovoří. O svém způsobu hry nebo o důvodech svého tahu nám bohužel počítač nic neřekne. Koncovku KD-KV sice perfektně ovládá, ale ne na základě zvláštní strategie nebo heuristiky. Musíme si uvědomit, že veškeré znalosti programu tvoří seznam o 1,9 milionech pozic, který si můžeme představit asi takto:

8. Počítačový šach pro profesionály

279

bKd4, bDg2, cKg4, cVg3 (32)
 bKf2, bDh4, cKh2, cVa1 (0)
 bKa8, bDa5, cKh6, cVe8 (61)
 bKe3, bDh5, cKg3, cVg2 (12)
 bKc5, bDh5, cKh4, cVa3 (39)
 bKe2, bDg4, cKh2, cVh1 (3)
 bKd8, bDa8, cKh6, cVe5 (51)
 ...
 ...
 atd. - celkem 1,9 miliónu záznamů

(číslo v závorkách uvádí, kolik pátahů zbývá do výhry. Sudá čísla znamenají černého na tahu, lichá čísla bílého na tahu).

Tento seznam používá počítač v praktické hře pro hledání tahu. Když vede bílé kameny (krále a dámu), vyhledává v seznamu příslušnou pozici a přečte číslo, které je u ní uvedeno. Pak generuje všechny pozice, které z této pozice vzniknou po jednom tahu, a také ty hledá v seznamu. Přinejmenším jednu z nich doprovází číslo, které je o jedno menší než číslo aktuální pozice. Počítač zahrne právě tento tah, který vede k této pozici. Když hraje černými kameny, pak naopak hledá pozice s nejvyššími čísly, aby kladl odpor co nejdříve a případně se i dostal za hranici 50 tahu.

To znamená, že počítač hraje koncovku *absolutně perfektně* (Ken Thompson: "Tak dobře jako bůh"). V roce 1988 dokonce ani Gari Kasparov a Anatolij Karpov, dva nejlepší hráči světa, nedokázali na turnaji v Holandsku společně proti počítači prosadit vítěznou pozici.¹⁴ Před nemilosrdnou precizností úplně analýzy musí většina lidí kapitulovat.

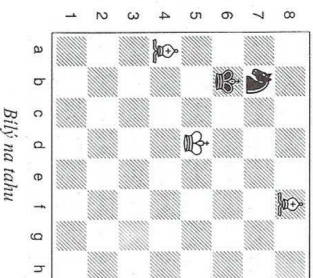
Koncovky s pěti kameny

Koncovka dány proti věži byla obtížná, ale pro lidi pochopitelná. To, co přišlo potom, dohnalo experty k zoufalství. Ken Thompson totiž začal analyzovat pětkamennou koncovku, což je zhruba stokrát složitější než analýza KD-KV. První zvo-
 lil koncovku dvou střelců proti jezdcí. Musel přitom vzít na vědomí novou dimenzi. Koncovka KD-KV vyžadovala prozkoumání 1,9 miliónu pozic, v případě koncovky

14 O zajímavém zážitku se čtenáři nejvíce velemistr Peter Leko hovoří. Předtím, při jedné návštěvě se Peter hodnou pokoušel vyhrát s dámu proti věži resp. hrát si s věží. Výsledek byl podle očekávání zklamavý. Na druhý den ale šachové zážitky už oznamoval, že už tuto koncovku pochopil - a dokázal to jasným vítězstvím proti počítači. Peter v noci před tím ležel dvě hodiny v posteli a vymýšlel strategii vítězství!

KSS-KJ už jich bylo 121 milionů! Aby se takové ohromné množství pozic dalo vygenerovat, prozkoumat a uvést do vzájemného vztahu, musel velký počítač pracovat téměř 700 hodin. Kromě toho musela mít operacní paměť takového počítače kapacitu 2,5 GB. Když byla analýza hotová, zažil šachový svět malý šok.

Koncovka dvou střelců proti jezdcí měla svoji historii. Před více než sto lety začala diskuse o tom, zda mohou dva střelci dosáhnout vítězství. Výsledek zkoumání mnoha generací teoretiků koncovek zní: když je jezdec velmí vřazen od krále a v důsledku toho jím nemůže být kryt, pak jej většinou dva střelci porazí. Jinak se silnější straně podaří udržet remizu. Příslušnou strategii popsal v roce 1851 teoretikové německého původu Kling a Horowitz. Její podstata spočívá v opevnění pro krále a jezdce:



Bílý na tahu

V knize *Praxe koncovek v šachu* (Endspielpraxis im Schach, Heyne 1977) uvádí velmistr Luděk Pachman "zanučené bezpečnou pozici" a píše: "Zde se černému nemůže nic stát, jeho král prosí pendulující na polích b6 a c7 a jezdec se z b7 vzdává, je vždy jen na okamžik, když může dát šach. Tato obranná pozice se nedá protáhnout a měli bychom o ni (máme-li jezdce proti dvěma střelcům) vždy usilovat - tedy umístit jezdce na jednom z polí b7, g7, b2, g2 a krále držet u něj." Výrok moderní teorie koncovek zní: "Koncovka KSS-KJ titoční vyhrává, pokud ovšem obě strany nezaujmou Kling-Horowitzovu pozici". Počítací dávká důkaz: koncovka je pro titočníka zásadně vyhraná, i když obránce zaujme Kling-Horowitzovu pozici.¹⁵ Maximální délka vítězné partie je 66 tahů.

¹⁵ Černý sice může dosáhnout Kling-Horowitzovy pozice, ale bílý jej odhánění tahy z kouta vyžene. Pak přetáhá řadu s mnoha nepřehlednými tahy, jímž bílý má všechny snahy černého o zaujetí Kling-Horowitzovy pozice v jiném rohu. Černý král je společen s jezdce zatlacen do rohu a jezdec je nakonec ztracen, ať už je blízko krále nebo ne.

Tento výrok vybyroval šachové profesionály a odborníky na koncovky po celém světě. Výsledek zjištěný počítačem byl nejen v rozporu s přijímanou teorií, ale vyžadoval dokonce změnu šachového pravidla ohledně remizy po 50 tazích. Thompson dokázal, že existují koncovky, které už jen teoreticky vyžadují pro vítězství vedení více než 50 tahů.¹⁶

Mnoho odborníků na koncovky zpočátku Thompsonovy analýzy zpochybňovalo. Americký velmistr Pal Benko varoval v časopisu *Chess Life* před slepou důvěrou k počítačům a kriticky rozebral Thompsonem zveřejněné partie. Byl toho názoru, že některé obranné tahy jsou chybné, protože mohl bílý v remizové pozici vyhrát. Když byly ovšem počítači předloženy jeho "správné" alternativní tahy, byla výhra bílého ještě rychlejší. Benko musel v dalším čísle *Chess Life* připustit, že počítač podcenil a přehlédl některá "zcela neuvěřitelná" pokračování.

V následujících deseti letech analyzoval Ken Thompson v Bell Laboratories v New Jersey všechny důležité pětkamenové koncovky. Na jednu koncovku potřeboval velký počítač asi dva až tři týdny celodenních výpočtů (to ovšem platilo jen pro koncovky s figurami, v nichž chyběly pěšci - viz níže). Zhruba 121 milionů pozic, které byly při jedné takové koncovce vytvořeny a prozkoumány, zabralo na pevném disku v průměru kolem 175 MB. To bylo 1,5 bytu na pozici (s oceněním), což na první pohled neje. Důvodem pro relativně malý soubor je to, že se díky otáčení a zrcadlení šachovnice podařilo eliminovat mnoho symetrií, to znamená, že se velký počet pozic dal převést na jiné, které tak bylo možné uložit jen jednou.

V následující tabulce je uvedeno několik koncovek s figurami, s údajem o četnosti výhry a maximální délce vítězné pozice.

Bílý	Černý	Max	Výhra	Bílý	Černý	Max	Výhra
KSS	KJ	66	91,8 %	KVV	KV	31	94,3 %
KD	KJ	63	89,7 %	KVD	KV	35	95,9 %
KD	KJ	42	93,1 %	KDJ	KD	35	48,4 %
KD	KJ	71	92,1 %	KDS	KD	33	53,4 %
KVJ	KV	33	35,9 %	KDV	KD	67	92,1 %
KVS	KV	59	40,1 %	KDD	KD	30	94,0 %

V případě prvních šesti koncovek (levý sloupec) je uveden počet tahů, které jsou potřebné k přechodu do triviální vyhrané koncovky. Tak například v pozici bK,a8, Dd1, eKd7, Sc5, Sd5 je při optimální hře obou stran potřeba 71 tahů k dobytí černého střelce, načež se dá koncovka KD-KS samozřejmě snadno vyhrát. V koncovkách v pravém sloupci jsou uvedeny nejdelší varianty až k matu. Pro každou koncovku je

¹⁶ Doposud bylo známo jen to, že existuje několik málo pozic se dvěma jezdci proti jednomu pěšci, které potřeby více než 50 tahů.

také uveden počet vítězných pozic v procentech ze všech možných pozic. Tak například v koncovce KD-KSS, kterou teorie považuje za remizu, podle Thompsonových výpočtů vede 92,1 % ze všech pozic k výhře bílého.

Pěšec porušuje symetrii

Při čisté figurových koncovkách jsme měli co do činění se 121 milióny pozic. Jakmile se ale mezi pěti kameny v koncovce objeví pěšec, vzroste dramaticky počet různých pozic, protože jednosměrný pohyb pěšce bourá mnohé symetrie. Při koncovce "dáma a pěšec proti dámě" existuje více než 335 miliónů pozic. Thompson na tuto koncovku se svým velkým počítačem potřeboval celkem šest týdnů. Přitom se nepočítá čas, který by byl potřebný k analýze všech koncovek, které vzniknou následkem přeměny pěšce (KDD-KD, KDY-KD, KDL-KD, KDS-KD).

Podle nastavení bílého pěšce může maximální počet tahů, které jsou potřebné k dalšímu tahu pěšce, daleko překročit číslo 50. Například s pěšcem na a6 (nebo h6) existují pozice, v nichž bílý vynutí krok pěšce až po 71. prezích tazích; naproti tomu je možné s pěšcem na a2 (h2) i při nepříznivé konstelaci figur dosáhnout pokroku v 17 tazích. Podíl pozic, které vedou k výhře, se pohybuje mezi 48,5 % (pro bílého pěšce na a3) a asi 85 % (bílý pěšec v sedmé řadě).

K výšec nejčastěji koncovkám s pěti kameny patří "dáma a pěšec proti dámě". Prověření známých teoretických knih za pomoci počítače ukazuje, že žádný z uznaných odborníků této koncovce nerozuměl. Všechny dosavadní práce byly plné chyb. Nasvědčují se tato koncovka příliš často nevyskytuje. V praxi je mnohem důležitější koncovka "věž a pěšec proti věži" (v koncovkové bibli Chérona je jí věnováno 124 stran). Počítač ukazuje, že nejdříve posloupnost zde obnáší 35 tahů (v pozici bKd1, Vd6, Pb2, cKg4, Vt7 je nutných 35 přesných tahů, než může být posunut pěšec). Podíl vítězných pozic kolísá od 89,1 % pro cPd7 do 44 % pro cPa2. Také v tomto případě je na tradiční teorii co napravovat.

Koncovky na CD

Koncovky Kena Thompsona byly senzací příležitostí pro experty, ale také prakticky turnajový hráč by dal mnohé za to, aby získal přístup k některým pětkamenovým řešením. To ale nebylo možné, protože databáze byly umístěny na velkých optických discích v Bell Laboratories. Několik specialistů na koncovky navštívilo Thompsona,

aby databáze prostudovali a napsali o nich články, ale bezprostřední kontakt s ní byl většinou hráčů odepřen.

V roce 1991 se Thompson rozhodl, že svoji práci zprístupní široké veřejnosti. Učinou technickou zkomprimoval databáze do té míry, že se daly uložit na běžný CD. Na třech CD-ROM s kapacitou po 600 MB se tak stěsalo celkem 66 pětkamenových koncovek. V roce 1995 se objevili čtvrtý disk, který doplnil nabídku, takže jsou dnes dostupné všechny důležité koncovky s pěti kameny.¹⁷

Pro ovládnutí databázi koncovek napsal Thompson velmi jednoduchý program v jazyce C, který umožňuje přístup k datům na CD. Daleko pohodlnější je napojení databáze na ChessBase pro Windows. Tato funkce je obsažena v demo verzi na doprovodném CD k této knize, takže můžete hned začít s databází koncovek experimentovat. Jedna se o koncovku král a dva střelci proti králi a jezdců, která byla na CD přibalená jako ochutnávka. Příslušný soubor má velikost 28 MB. Všechny ostatní pětkamenové koncovky jsou k dostání v odborných prodejnách za celkovou cenu kolem 250 DEM.

Chcete-li experimentovat s modulem pro analýzu koncovek, spusťte ChessBase a načtěte z databáze partii nebo pozici. Partie obsahující koncovku můžete najít velmi rychle pomocí vyhledávací funkce: v seznamu partií stisknete (jak bylo popsáno výše) kombinaci kláves Ctl + M a ve vyhledávací masce kliknete na tlačítko "Material". U bílého střelce nastavíte hodnoty "2-2" (= nejméně a nejvíce dva střelci) a u černého jezdce "1-1". Pěšce musíte nastavit na "0-0". Ujistěte se, že je aktivována volba "Farben ignorieren" ("Ignorovat barvy") - chceme totiž najít také partie, v nichž bojovali dva černí střelci proti bílému jezdců.

V databázi *Mainbase* najdete jedinou partii, v níž se vyskytla tato koncovka: Borvinnik-Tal, MS Moskva 1961. V 77. tahu přešel Borvinnik do koncovky a pokusil se zaujmout Kling-Horowitzovu pozici. Tal tomu ovšem zabránil a rychle vyhrál. Tuto partii můžete načíst a pozici analyzovat pomocí databáze koncovek.

Další příklady koncovky "dva střelci proti jezdců" najdete v databázi *Computer*.

Analýtický modul se aktivuje kombinací kláves Alt + F5. Objeví se seznam se všemi tahy, jež jsou v dané pozici přípustné. Za každým tahem je uveden stav: kladné číslo znamená, že tah vyhrává z hlediska bílého (při optimální obraně protivníka) v uvedeném počtu tahů. Nula znamená, že tento tah vede k remíze. A konečně záporné číslo označuje výhru černého. Většinou existuje jedna nebo dvě dobré možnosti,

¹⁷ Čtyřkamenové koncovky jsou obsaženy v pětkamenových. Žádá databáze ovšem neobsahuje víc než jednoho pěšce, takže např. koncovka dva pěšci proti jednomu pěšci není k dispozici. Pro Thompsona je tento typ koncovky zřím směr. Nejsou zde žádné symetrie a je nutné zohlednit 75 "poklonovek" (DP-P, VP-P, JP-P atd.)

mnoho remizových variant a několik prohrávajících tahů. Pomocí mezeru lze přenést tah přenesení na šachovnici. Když tedy budete neustále tisknout tuto klávesu, vytvoříte automaticky optimální hlavní variantu.

Tažší v tvář složitosti koncovky analyzovaných Thompsonem složila většina odborníků zbraně. Jen nemohli se seřizovat pokoušet využít databázi k tomu, aby definovali strategii, které by měly smysl pro člověka. Jednou výjimkou je anglický velmistr a analytik Dr. John Nunn, který doposud zveřejnil tři vynikající didaktické knihy o pětkamenných koncovkách.¹⁸ André Schulz, spolupracovník firmy ChessBase, vede státní rubriku v elektronickém časopise *ChessBase Magazin*. Třetí aktuální turnajové partie a za pomoci databáze zkoumá jejich koncovky. Pečlivé studium těchto příkladů přinese mnoho i praktickým hráčům.

Mnoho špičkových hráčů si koupilo notebooky s jednotkami CD-ROM a berou si CD s koncovkami "pro všechny případy" s sebou na turnaje. Plně doma cvičí, nejen proto, že chtějí posbírat kdejakou firmu. Mimo to totiž při práci s databází koncovek vidí tahy, které jakoby podporují všem strategickým pravidlům šachu - například střelce, kteří jsou v boji proti jezdcům v takové harmonii, s jakou se v praktické hře ještě nikdo nesešel. Takové tahy a manévry se pak snaží použít i v jiných fázích partie.

Jak daleko se dá jít?

Pětkamenné koncovky jsou prakticky vyřízeny, jejich úplná analýza je na CD. A co se šesti kameny? V roce 1991 se skutečně našel vědec jménem Lewis Stiller z Baltimore (USA), který se odvažil pustit do první šestikamenné koncovky. Použil k tomu počítač se 65536 paralelními procesory (ano, pětadesát tisíc centrálních jednotek!) a Thompsonovou metodu zkoumal koncovku věz a střelce proti dvěma jezdcům. Počítač vygeneroval a ocenil 100 miliard legálních pozic s následujícím výsledkem (nejedná se o žert): věz a střelec mohou vyhrát, a sice nepozději po 223 tazích! Skutečně existují pozice (např. bKa7, Vb2, Sb3, tKd3, Jc6, Jd6) v nichž musí útočník při optimální hře protivníka zahrát 223 přesných tahů, aby dobytí jezdeck. Odborníci, kteří si tuto poslovnost tahů přehráli, v ní nenašli žádnou pro lidi využitelnou strategii.¹⁹

¹⁸ V anglickém nakladatelství Batsford. Jsou to první šachové knihy, které zaručeně neobsahují žádné chyby.

¹⁹ Je jen vidět, jak blízko asi 200 tahů má být manévry bez jasného cíle. Černý král je postupně vytlačen na okraj, jezdcí mají stále větší problémy, jak se navzájem ubránit. Nakonec je jeden z nich ztracen.

Tato databáze obsadí ve Stillerově paralelním počítači 8 GB paměti RAM. Procesory se o ni totiž dělí velmi komplikovaným způsobem, takže se její obsah nemůže přehrávat na magnetickou pásku nebo optický disk. Když se počítač vypne, je po všem. Můžeme ale počítat s tím, že v blízké budoucnosti budou k dispozici také šestikamenné koncovky. Ken Thompson už začal první z nich analyzovat na "normálním" počítači a připravovat techniku, která umožní zápis gigantických seznamů na datové nosiče. Za pomoci dokonalejších kompresních metod se snad tyto koncovky dostanou na CD nové generace.²⁰

Jak daleko můžeme - nebo máme - jít? Dánský velmistr Curt Hansen byl pobouřen existencí databáze a knih Johna Nunn. "Byl jsem šachem vždy fascinován", píše, "protože tato hra navozuje velké množství zajímavých problémů, které je tak obtížné vyřešit, že na ně nikdo nemůže najít přesnou odpověď. Přesto mají šachisté různých úrovní tu a tam pocit, že sehráli fantastickou partii." Analýzy koncovek ale navždy zbavily kouzla celou jednu oblast šachu. "Má budoucnost šachu vypadat tak, že se hráči už jen budou nazpaměť učit odpovědi na exaktní otázky? Je správné, že někteří velmistré tento vývoj ve svých knihách urychlují?" Hansen doporučil, že by nadrodní šachové federace měly firmám jako je IBM psát protestní dopisy, protože - také díky počítačům typu Deep Thought - přispívají ke zničení šachové hry.

Ve své odpovědi poukázal John Nunn na to, že se nikdo nemůže naučit nazpaměť dvě miliardy bytů informací. "Kdyby ale bylo tyto megabyty možné vyjádřit několika jednoduchými pravidly, kterým by rozuměli lidé, pak by se znalost o pětkamenných koncovkách podstatně rozšířila. Když jsem začal pracovat na své knize o koncovkách s věžemi, bylo velkou otázkou, zda lidé porozumí výsledkům z databáze. Nebo tato data zůstanou jen seznamem náhodných pozic z nichž nebudeme moci odvodit žádné všeobecné principy? Můj závěr zní, že ve většině případů je lidské porozumění možné, ačkoli pochybují o tom, zda kdy plně pochopíme koncovku dána a pěšce proti dámě."

Podle Nunn budou důsledky takovéto: V budoucnosti už nebudou žádné významné partie (ale trend je i bezak takový). Na turnajích se bude muset hrát zakázat jakákoli komunikace s jinými osobami (vítězný tah by mohl zjistit počítač v jiné části budovy). A konečně - hráči začnou omezovat ostrá zahájení (protože by protivníci za pomoci počítače mohli najít příliš mnoho taktických pastí).

"Předpokládáme", píše Nunn, "že by vznikla databáze všech legálních pozic šachové hry. Zdáte výchozí pozici a počítač řekne, že 1.h4 prohává po 231 tazích,

²⁰ Další generace CD má mít kapacitu "pouličit" 2 resp. 7 GB, zatímco šestikamenné koncovky potřebují asi 5 až 10 GB paměti. Ale zdá se, že v budoucnu, ve vývoji jsou holografické CD s kapacitami udávajícími řádově v TB (1 TB = 1000 GB). To je asi 700 řádů v TV kvalitě nebo velké množství šesti-kamenných koncovek, které nebudou ani potřebovat kompresi.

I.f3 po 359 tazích a všechny ostatní tahy zaznamenávají remízu. Je to konec šachové hry? Já věřím, že ne. Sportovní stránka šachu by zůstala téměř nezměněna, jen vědcům by se otevřelo zcela nové pole působnosti: jak interpretovat informace obsažené v databázi.”

I když vás slova Johna Numma neuklidňila, zbývá vám jedna útěcha: úplný propočít 32kamenové koncovky se zřejmě v dějinách tohoto vesmíru uskutečnit nepodaří.