Katedra Informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Analýza TETRISu ako hry dvoch hráčov

Martin Kravec

BAKALÁRSKA PRÁCA

9.2.1 Informatika

Vedúci: Mgr. Ľuboš Steskal

Bratislava, 2009

Abstrakt

V tejto práci formulujeme tetris ako hru dvoch hráčov. Popisujeme algoritmus pre hranie za oboch hráčov a snažíme sa nájsť k nemu prislúchajúcu heuristickú funkciu. Hľadáme výherné stratégie a ukazujeme, že v závislosti od obmedzenia pravidiel existujú výherné stratégie pre hráča, ktorý ukladá kocky, ale aj pre hráča, ktorý ich hádže. Práve pre hráča, ktorý hádže kocky ponúkame všeobecnú výhernú stratégiu s časovou zložitosťou $O(n \cdot m)$ na jeden ťah, kde n je šírka a m výška hracieho poľa.

Kľúčové slová: tetris, stratégia, heuristiká funkcia.

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím citovaných zdrojov.

. . . .

Poďakovanie

Rád by som sa poďakoval hlavne svojmu vedúcemu Mgr. Ľubošovi Steskalovi, ktorý sa mi po celý čas venoval a mal so mnou strpenie.

Ďalej by som sa chcel poďakovať mojej priateľke Lenke Matejovičovej za morálnu podporu a za pomoc pri gramatickej korektúre.

Moja vďaka patrí aj kamarátom, ktorý mi pomohli prekonávať prekážky pri tvorbe tejto práce v LaTeXu.

Obsah

Zoznam obrázkov

Úvod

TETRIS je počítačová hra, ktorú 6. júna 1984 vytvoril Alexej Pažitnov, keď pracoval na moskovskej akadémii vied, inšpirovaný stolnou hrou pentomino[1]. TETRIS je najslávnejšou počítačovou hrou celej histórie, čiastočne vďaka svojmu úspechu vo forme videohry. Neskôr sa spolu s rôznymi verziami, známymi ako klony, dostal takmer do všetkých elektronických zariadení obsahujúcich dostatočne veľkú obrazovku. V nasledujúcej práci sme sa rozhodli skúmať niektoré zaujímavé vlastnosti tejto hry. Definovali sme si ju ako hru pre dvoch hráčov, toho klasického, ktorý ukladá kocky ale aj toho, ktorý ich hádže. Analyzujeme rôzne stratégie, predkladáme časovo efektívny algoritmus schopný hrať hru na dostatočne vysokej úrovni a skúmame teoretické hranice optimálneho správanie pre oboch hráčov. Práca z veľkej miery čerpá z diela Johna Brzustowského - „Can you win at Tetris?“[2], jedného z prvých priekopníkov tetrisu v teoretickej oblasti. Obrázky 2.1 až 4.3 sú tiež prevzaté z tohoto článku.

Kapitola 1

TETRIS

V nasledujúcej práci sa budeme venovať oficiálnej dvojrozmernej verzii počítačovej hry TETRIS.

1.1 Definícia hry TETRIS

Definujeme hru TETRIS podľa jej oficiálnych pravidiel originálnej verzie[3].

V originálnej verzii tetrisu má hracie pole veľkosť 10×20 štvorčekov (obrázok 1.2).

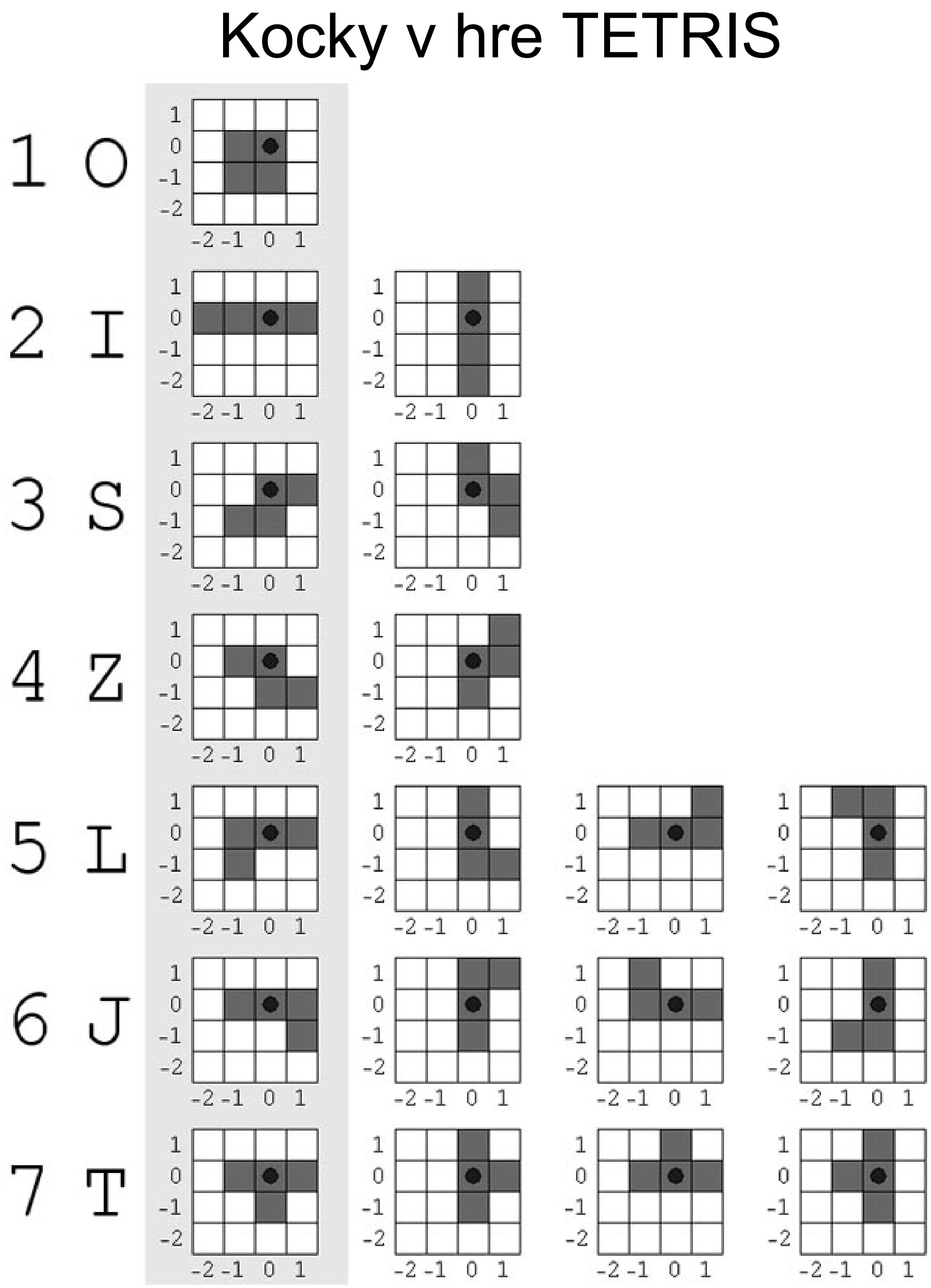
Hráč má na výber zo 7 tetromín (hracích kociek) (obrázok 1.1), zložených práve zo štyroch štvorčekov. Kocky môžu mať rôzne mená. My si ich pomenujeme ako **O, I, S, Z, L, J** a **T**.

Pohyby kocky v hre:

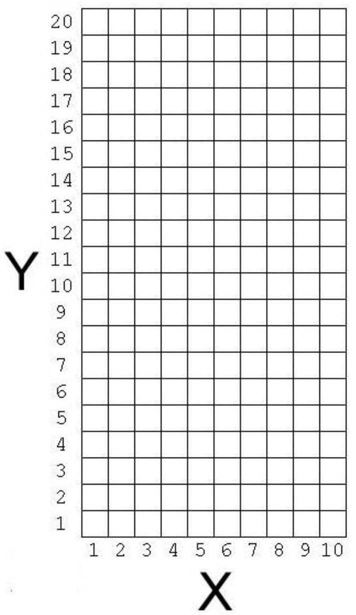
- posun o jeden štvorček doľava, doprava, alebo dole
- otočenie o 90° v smere, resp. proti smeru hodinových ručičiek
- spadnutie na najnižšie možné umiestnenie, teda najnižšie možné povolené umiestnenie
- automatické posunutie kocky smerom dole o jeden štvorček v časových intervaloch určených levelom (v štandardnej verzii bolo 10 levelov)

Nepovolený krok tetromina nastáva:

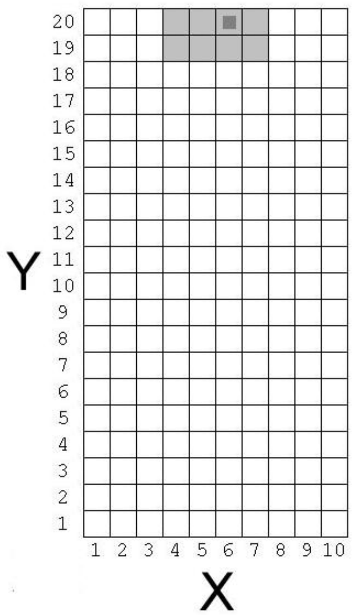
- ak sa ktorýkoľvek zo 4 štvorčekov kocky dostane mimo hracieho poľa
- ak je ktorýkoľvek zo 4 štvorčekov kocky umiestený na štvorčeku na hracom poli, ktorý už bol vyplnený predtým



Obr. 1.1: Kocky / Tetrominoes



Obr. 1.2: Prázdne hracie pole



Obr. 1.3: Miesto pre zrod novej kocky

Povolený krok je každý krok, ktorý vznikne buď automatickým pádom o jeden štvorček, alebo pohybom kocky v hre a nie je to nepovolený krok.

Tetromino je umiestnené do hracieho poľa, ak by posun o jedno políčko nižšie už nebol povolený krok. Ak je tetromino umiestnené do hracieho poľa, štvorčeky v hracom poli obsahujúce novú kocku sú vyplnené a s tetrominom sa už nedá hýbať. Nastáva ďalší ťah.

Ak má po umiestení kocky do hracieho plánu niektorý riadok vyplnené všetky štvorčeky, tento riadok je zrušený, hráčovi sú pripísané body, všetky riadky nad zrušeným riadkom sú posunuté o jeden riadok nižšie a na úplný vrch hracieho plánu je doplnený nový riadok bez vyplnených štvorčekov.

Všetky tetrominá sa rodia na rozlohe 4x2 štvorčeky (obrázok 1.3) a tým začína ťah hráča. Ak narodením tetromina nastane nepovolený krok, hra je ukončená.

Originálna hra má aj špeciálne bodovanie odvíjajúce sa od levelu a počtu riadkov zrušených jednou kockou, ale v tejto práci sú body bezpredmetné, takže sa nimi nebudeme zaoberať.

1.2 Motivácia

Už od vzniku TETRISu a ustálenia oficiálnych pravidiel bolo nájdenie stratégie na výhru, či prehru pre matematikov výzvou. Preto sa touto myšlienkou zaoberali a celú teóriu sa snažili zapísať formálne. Momentálne existuje mnoho variánt teórií TETRISu.

Zoberme si napríklad offline verziu TETRISu. Offline (full-information) verzia je taká, v ktorej vieme dopredu presnú konečnú postupnosť kociek. Demaine, Hohenberger a Liben-Nowell[4] ukázali, že v takejto teórii je NP-úplný problém optimalizovanie nasledujúcich funkcií (niekoľko prirodzených otázok, ktoré hráča napadnú pri hraní tetrisu):

- maximalizovanie počtu zmazaných riadkov kým nie je koniec postupnosti kociek
- maximalizovanie počtu kociek kým nenastane prehra hráča
- maximalizovanie počtu tetrisov - štvoríc riadkov zmazaných v jednom ťahu
- minimalizovanie výšky najvyššieho zaplneného štvorčeka v hracom pláne počas celej postupnosti kociek

1.3 Algoritmus

NP-úplné problémy sú ťažko riešiteľné, preto sme sa rozhodli pristúpiť k hre z iného hľadiska a nájsť výpočtovo efektívnejší algoritmus. Budeme simulovať reálnu hru s mierne upravenými pravidlami.

Majme dvoch hráčov. Jeden bude klasický hráč, ktorý dostane nejakú kocku a tú sa bude snažiť niekde uložiť, pre neho čo najvýhodnejšie. Jeho snahou bude hrať hru večne, vtedy vyhrá nad druhým hráčom. Druhý hráč bude vyberať pre prvého hráča kocky. Vyhrá vtedy, keď prvý hráč prehrá. Jeho cieľom bude teda prinútiť prvého hráča ukončiť hru po konečnom počte ťahov. Nazvime si prvého hráča ako Dobrého a druhého hráča ako Zlého - zlého preto, lebo sa snaží ukončiť hru (aby prvý hráč prehral). Zadefinujme si niekoľko základných pojmov, s ktorými budeme v celej práci pracovať.

Definícia 1.1. *Výška stĺpca* je výška najvyššieho zaplneného štvorčeka v stĺpci.

Definícia 1.2. *Diera* je nezaplnený štvorček v hracom pláne, ktorý má nad sebou alebo po oboch stranách zaplnený štvorček. Definovali sme 3 typy dier:

Obr. 1.4: Príklady troch typov dier.

- Veľká diera - prázdny štvorček, ktorý má nad sebou a po oboch stranách vedľa seba zaplnený štvorček.
- Stredná diera - prázdny štvorček, ktorý má nad sebou zaplnený štvorček ale nie je to veľká diera.
- Malá diera - prázdny štvorček, ktorý má po oboch stranách vedľa seba zaplnený štvorček, ale nie je to veľká diera.

Príklady týchto dier môžeme vidieť na obrázku 1.4.

Ako výpočtový algoritmus sme si vybrali Min-Max a upravili sme ho pre svoje potreby. Prvý (Dobrý) hráč bude Min, druhý (Zlý) bude Max.

Algoritmus funguje nasledovne:

- Zlý hráč (Max) má k dispozícii množinu typov kociek (tetromín). Postupne vyberá každú kocku z množiny a pozrie sa, ako by ju uložil Dobrý hráč. Ten mu vráti hodnotu najlepšieho uloženia kocky do hracieho plánu, teda najnižšiu hodnotu plánu akú vie s danou kockou spraviť. Zlý hráč si nakoniec vyberie tú kocku, ktorá mu zaručí najvyššie ohodnotenie plánu a hodnotu vráti späť.
- Dobrý hráč (Min) dostane vo svojom ťahu od Zlého hráča kocku. Vygeneruje všetky možné otočenia a umiestnenia kocky do hracieho plánu a pre každé takéto umiestnenie:
  - ak sa ešte pozerá dopredu, tak zistí, aká kocka by nasledovala v ďalšom ťahu (teda sa pozrie, ako by hral Zlý hráč), a nakoniec vráti najmenšiu hodnotu, ktorú vráti Zlý hráč zo všetkých umiestnení a tiež vráti konkrétne umiestenie.

  – ak sa už nepozerá dopredu, tak ohodnotí Ohodnocovacou funkciou vzniknutý hrací plán a späť vráti najmenšiu hodnotu a jej prislúchajúce umiestnenie zo všetkých možných umiestnení.

Tento algoritmus generuje výpočtový strom ako na obrázku 1.5.

Dôležitá je teda Ohodnocovacia funkcia, ktorá určí, ako je konkrétny hrací plán dobrý. Čím nižšia je hodnota hracieho plánu, tým je plán výhodnejší pre Dobrého hráča a čím vyššia je hodnota hracieho plánu, tým je plán výhodnejší pre Zlého hráča.
Túto funkciu sme si postupne vytvárali sami. Vychádzali sme z niekoľkých parametrov, ktoré sme odsledovali pri skúmaní hry a ktoré sme považovali za podstatné:

- maximálna výška stĺpca v hracom pláne
- priemerná výška stĺpcov v hracom pláne
- množstvo rôznych dier, ktoré mali rôzne stupne vážnosti

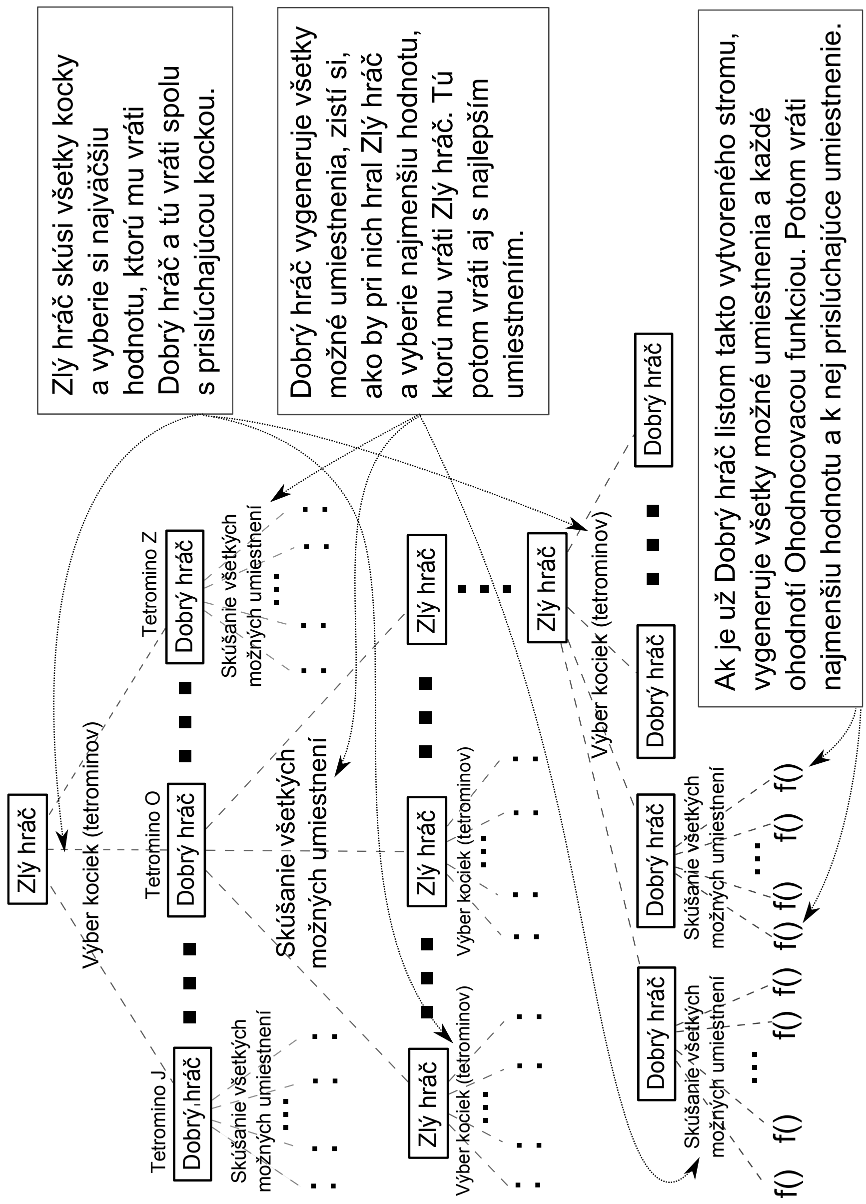
V ďalších kapitolách sa budeme venovať vytváraniu tejto funkcie.

1.3.1 Generovanie možných umiestnení

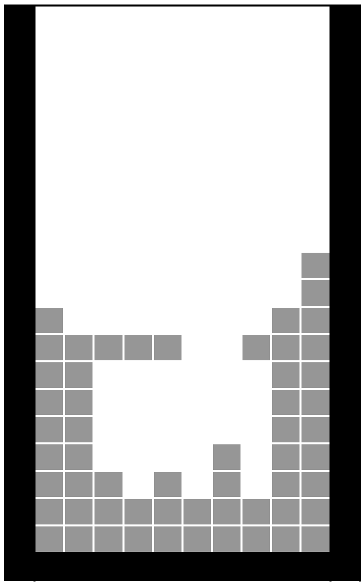
Jednou z úloh, ktoré musí robiť Dobrý hráč vo svojom ťahu, je aj generovanie všetkých možných umiestnení.

Ako prvé si vytvorí všetky otočenia danej kocky. Niektoré kocky majú len jedno otočenie (kocka typu **O**), niektoré majú dve otočenia (kocky typu **I**, **S** a **Z**) a ostatné majú štyri otočenia (kocky typu **L**, **J** a **T**). Pre každé otočenie zvlášť potom skúsi, aké sú možné uloženia do hracieho poľa, teda pri ukladaní už kocku neotáča. To by totiž znamenalo veľké zväčšenie výpočtovej náročnosti.

Problém by nastal, ak by túto skutočnosť vedel využiť Zlý hráč. Jedinou možnosťou na využitie je tvorenie tzv. veľkých jaskýň (obrázok 1.6), ktoré majú príliš malý otvor na to, aby ním mohli prejsť všetky kocky bez nutnosti správneho natočenia, pričom jaskyňa v sebe skrýva dostatok miesta na ľubovoľné uloženie kocky. Počas celej tvorby programu, testovania a následného používania sme sa ale ani raz nestretli s tým, aby sa Zlému hráčovi čo i len na chvíľu podarilo vytvoriť takúto jaskyňu a teda to ani nevyužíval. Dôvod bol ten, že Dobrý hráč vo svojej Ohodnocovacej funkcii považuje za dostatočne zlé vytvárať takéto jaskyne (napríklad minimalizovanie priemernej výšky a tiež minimalizovanie počtu dier, ktorých je v jaskyni veľa), a preto sa snaží vytvoreniu takýchto jaskýň zabrániť.



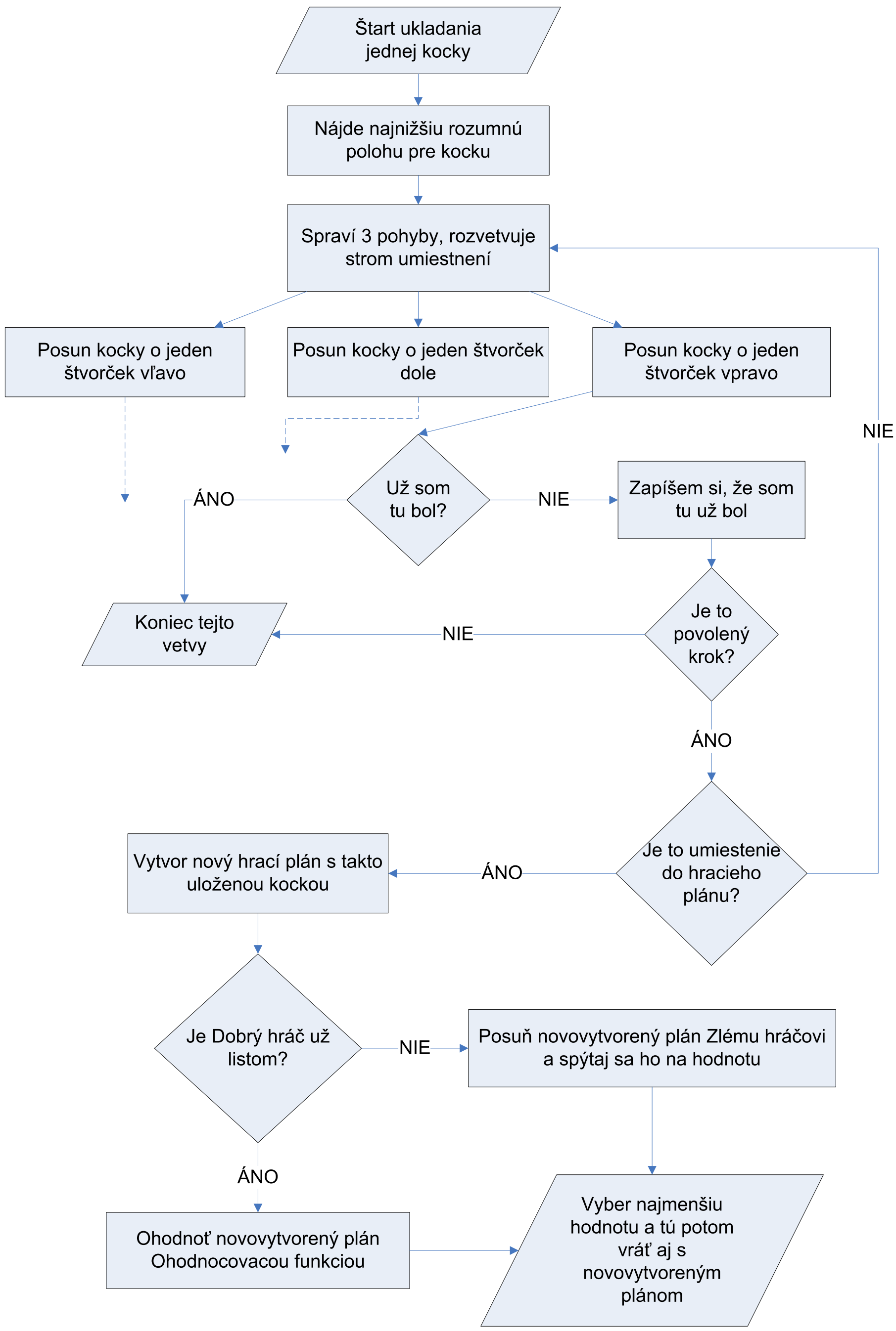
Obr. 1.5: Algoritmus programu



Obr. 1.6: Príklad veľkej jaskyne

Definícia 1.3. *Najnižšia rozumná poloha kocky* je taká poloha, z ktorej sa ešte stále dá uložiť kocka na všetky možné umiestnenia. To znamená, že jej najspodnejší štvorček musí byť vyššie ako štvorček v najvyššom stĺpci v hracom poli.

Algoritmus pre ukladanie jednej kocky (už otočenej) je popísaný na obrázku 1.7.



Obr. 1.7: Algoritmus ukladania jednej kocky

Kapitola 2

Stratégie

Ukázali sme, že jedna z kľúčových častí celého algoritmu, ktorým by sme radi simulovali hru, je Ohodnocovacia funkcia. O vhodnom tvare tejto funkcie zatiaľ veľa nevieme. Mala by vedieť **dobre** ohodnotiť hrací plán. Ohodnocovacia funkcia sa nazýva dobrá, ak si z dvoch hracích plánov vyberie ten, ktorý vedie k dlhšej, respektíve k nekonečnej hre. Čiže by mala subjektívne povedať, ktorý plán je lepší pre Dobrého hráča.

Definícia 2.1. *Dobrá* Ohodnocovacia funkcia je taká, ktorá si z dvoch hracích plánov vyberie ten, ktorý vedie k dlhšej, respektíve k nekonečnej hre.

Ako ale vieme povedať, ktorý plán je lepší? Ak by sme to nevedeli, môže to dokázať nami vytvorená funkcia? Ako ju teda budeme vytvárať?

Práve tieto otázky nás najviac trápili. Pri skúmaní tohoto problému sme sa presvedčili, že nie vždy je jednoduché povedať, ktorý plán je lepší, výhodnejší. Preto je nutné pristúpiť k tomuto problému trochu inak. Postupne budeme ukazovať a dokazovať rôzne vyherné stratégie, ale aj stratégie, ktoré donútia Dobrého hráča prehrať hru. Našou snahou bude vylepšovať Ohodnocovaciu funkciu tak, aby kopírovala tieto stratégie. Simulácia hry nemusí mať rovnakú postupnosť krokov ako dôkaz stratégie, ale musí mať rovnaký výsledok:

- ak má vyhrať Dobrý hráč, simulácia hry bude nekonečná.
- ak má vyhrať Zlý hráč, simulácia hry skončí v konečnom počte krokov prehrou Dobrého hráča.

Ak bude funkcia spĺňať tieto predpoklady, budeme ju považovať za **uspokojivú** Ohodnocovaciu funkciu.

Definícia 2.2. *Uspokojivou* Ohodnocovacou funkciou budeme volať každú funkciu, ktorej pozorované praktické výsledky (simulácie) sa zhodujú s teoretickými poznatkami (stratégie).

2.1 Úvod do stratégií

Predtým, ako sa pustíme do konkrétnych stratégií, si definujeme niekoľko základných pojmov s ktorými budeme pracovať tak, ako ich definoval J. Brzustowski vo svojej práci [2], v ktorej sa venoval práve nasledujúcim stratégiám.

Hrací plán veľkosti $n \times m$ má n sĺpcov číslovaných od 1 po n.

Komín budeme nazývať objekt zložený z dvoch susediacich stĺpcov. Ak sa nepovie inak, bude platiť, že ľavý z nich má nepárne číslo. Obvyklý hrací plán veľkosti 10×20 bude mať teda 5 komínov.

Definícia 2.3. *Komín* je dvojica susediacich stĺpcov hracieho poľa.

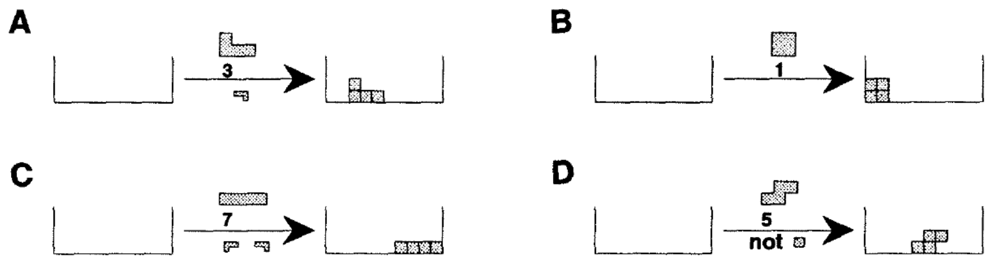
Číslo stĺpca, kam chceme uložiť kocku do hracieho plánu, bude číslo stĺpca, kam sa uloží jej najľavejší štvorček.

Pre jednoduchší zápis stratégií budeme používať šípkové diagramy popisujúce prechod z jedného hracieho plánu do druhého. Pred šípkou je hrací plán pred uložením kocky. Za šípkou je hrací plán po uložení kocky do stĺpca, ktorého číslo je zadané nad šípkou. Veľká kocka nad šípkou znázorňuje, ktorú kocku teraz máme uložiť do hracieho plánu. Pod šípkou môže byť malá kocka, ktorá príde v ďalšom ťahu (tzv. videnie jednej kocky dopredu - náhľad,lookahead), alebo je tam viac kociek ktoré eventuálne môžu prísť. Pod šípkou sa ale nemusí nachádzať nič (to znamená, že nemáme žiadnu dodatočnú informáciu o ďalšom priebehu hry), alebo sa tam môže nachadzať **not** malá kocka, čo znamená, že táto kocka v ďalšom ťahu určite nepríde. Pre čo najlepšie pochopenie je tento postup popísaný na obrázku 2.1.

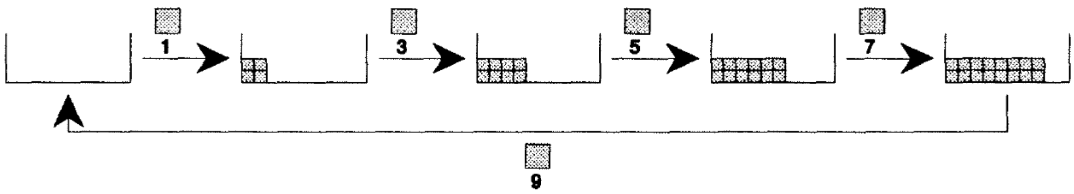
2.2 Základné stratégie

Na obrázku 2.1 vidieť iba fragmenty inštrukcií. Stratégiu budeme definovať práve takýmito inštrukciami. Ak pre konkrétnu hru dokážeme vytvoriť konečný počet inštrukcií, ktoré popisujú celú hru, vytvorili sme stratégiu.

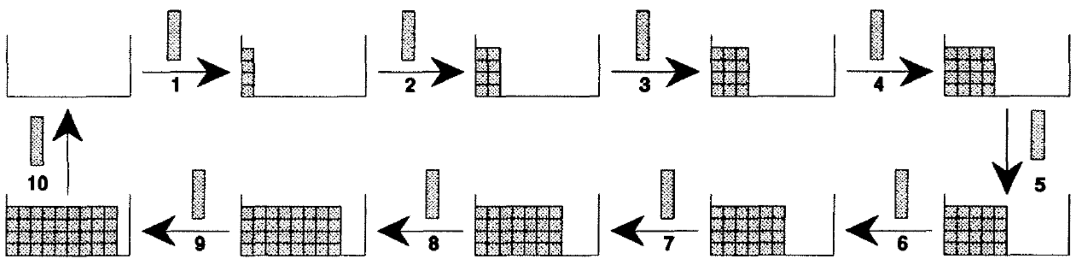
Zoberme si teda najjednoduchší prípad hry, v ktorej budeme mať klasické pravidlá hry, ale Zlý hráč bude mať k dispozícii len jednu kocku. Na obrázku 2.2 je znázornený jednoduchý príklad stratégie pre kocku typu O. Vidíme, že najvyšší zaplnený štvorček v ktoromkoľvek hracom pláne v postupnosti inštrukcií je v druhom riadku. Toto číslo budeme nazývať **výškou stratégie**. Ak teda bude kocka typu O jedinou kockou v hre tetris, toto bude výherná stratégia pre Dobrého hráča, keďže sa nikdy nezaplní štvorček na vyššom riadku ako 2.



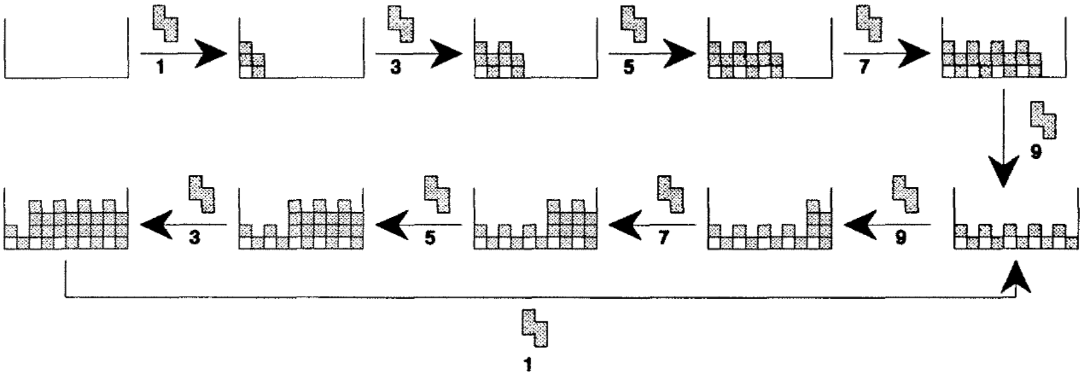
Obr. 2.1: 4 príklady šípkových diagramov a ich vysvetlenie. **A:** Ak sú znázornené obe kocky, práve ukladaná kocka (veľká) aj kocka v ďalšom ťahu (malá) typu L, uložíme kocku tak, že jej najľavejší štvorček bude v stĺpci 3. **B:** Ak je práve ukladaná kocka typu O a o kocke v ďalšom ťahu nevieme nič, uložíme kocku tak, že jej najľavejši štvorček bude v stĺpci 1. **C:** Ak práve ukladaná kocka je typu I a kocka v ďalšom ťahu môže byť typu L alebo J, uložíme kocku horizontálne tak, že jej najľavejší štvorček bude v stĺpci 7. **D:** Ak je práve ukladaná kocka typu S a kocka v ďalšom ťahu určite nebude typu O, uložíme ju tak, že jej najľavejší štvorček bude v stĺpci 5.



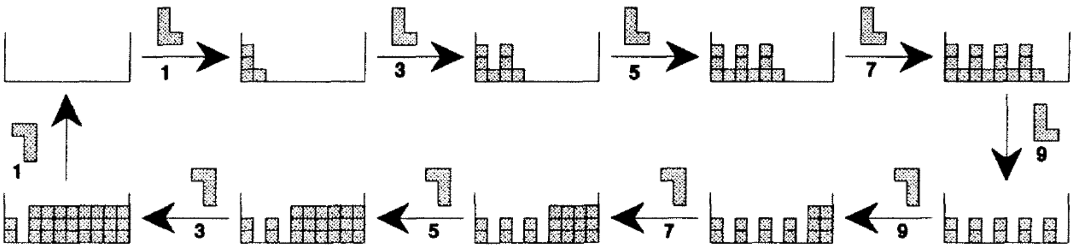
Obr. 2.2: Stratégia len pre kocku typu O. Zmaže 2 riadky po uložení kocky do 9. stĺpca a hra sa týmto ťahom vráti do pôvodného prázdneho hracieho plánu.



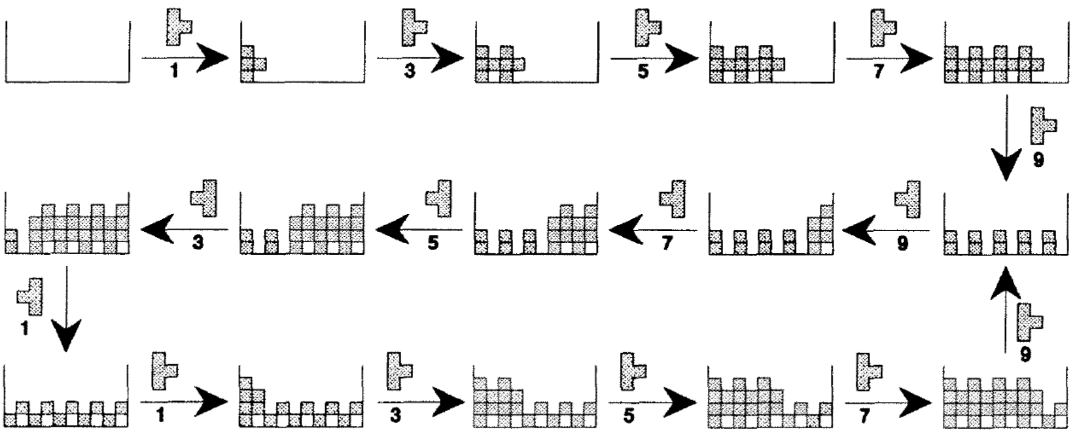
Obr. 2.3: Stratégia pre kocku typu I s **výškou stratégie** 4 a 10 stavmi/inštrukciami



Obr. 2.4: Stratégia pre kocku typu S s **výškou stratégie** 4 a 10 stavmi/inštrukciami



Obr. 2.5: Stratégia pre kocku typu L s **výškou stratégie** 3 a 10 stavmi/inštrukciami



Obr. 2.6: Stratégia pre kocku typu T s **výškou stratégie** 4 a 15 stavmi/inštrukciami

Definícia 2.4. *Výška stratégie* je výška najvyššieho zaplneného štvorčeka v ktoromkoľvek hracom pláne v postupnosti inštrukcií celej stratégie.

Kocka typu O nie je jediná, pre ktorú sa dá nájsť takáto „jednokocková“ výherná stratégia. Na obrázkoch 2.3 až 2.6 sú ukázané stratégie pre kocky typu I, S, L a T. Kocka Z je symetrická s kockou typu S, resp. kocka J je symetrická s kockou typu L, preto som ukázal len stratégiu pre jednu verziu týchto dvojíc kociek.

Ako vidíme na obrázkoch 2.2 až 2.6, dokázali sme vytvoriť stratégiu pre každú samostatnú kocku, keď má hracie pole šírku 10 štvorčekov. V klasickej hre TETRIS malo hracie pole práve takúto veľkosť. Samozrejme by nás mohli zaujímať aj iné šírky. Pokúsime sa teda ukázať, že tieto stratégie platia pre každú párnu šírku hracieho poľa.

Veta 2.1. *Pre každú párnu šírku hracieho poľa a ľubovoľnú kocku existuje výherná stratégia pre Dobrého hráča.*

Dôkaz. Pre dokázanie tohto tvrdenia použijeme už skôr definovaný pojem komínov. Predstavme si, že v každej stratégii je hracie pole rozdelené do 5 komínov, ktoré budeme číslovať zľava doprava počnúc jednotkou. To znamená, že komín 1 bude obsahovať stĺpec 1 a 2, komín 2 bude obsahovať stĺpec 3 a 4 atď. Všimnime si, že v každej stratégii je kocka uložená práve do jedného komína, teda štvorčeky kocky nie sú uložené v dvoch rôznych komínoch. Naviac v každej stratégii môžeme nájsť **vzor** (pattern) pre ukladanie kocky, ktorý je identický pre každý komín. Tento vzor môžeme opakovať pre ľubovoľný počet komínov a takto dostaneme výhernú stratégiu pre každú párnu šírku hracieho poľa. Týmto spôsobom sa nám v žiadnom prípade nezvýši výška stratégie. Zostane rovnaká ako pri hracom poli so šírkou 10 štvorčekov, s niekoľkými výnimkami. Pri šírke 2 má stratégia pre kocku typu O nulovú výšku, stratégia pre kocku typu I výšku 4 a stratégie pre ostatné typy kociek výšku 2. Špeciálnym prípadom je ešte kocka typu I, pre ktorú má stratégia pre šírku 4 výšku 0, a pre šírky 8, 12, 16, . . . výšku 1 (kocku budeme ukladať horizontálne). ■

A čo nepárna šírka hracieho poľa? Existuje aj tam pre každú kocku výherná stratégia? Problém nastane už pri kocke typu O.

Veta 2.2. *Pre kocku typu O neexistuje výherná stratégie pre Dobrého hráča pri nepárnej šírke hracieho poľa.*

Dôkaz. Kocka O zaberá pri každom otočení šírku dvoch štvorčekov a tým pádom nedokáže zaplniť (čiže ani zmazať) riadok s nepárnou šírkou. Ak sa nám zvyšuje počet štvorčekov uložených v hracom pláne, ktoré nemôžeme zmazať, vedie to k zaplneniu hracieho plánu a teda k prehre Dobrého hráča. ■

Situáciu trochu zachraňuje kocka typu I. Spôsob ukladania kocky z obrázku 2.3 dáva stratégiu pre kocku I aj pre nepárnu šírku hracieho plánu. Obrázky 2.7 a 2.8 ukazujú výherné stratégie pre kocky typu L a T pri šírke 3 a 5. Ak si v oboch stratégiách pozrieme vzor tvorby komína obsahujúceho stĺpce 4 a 5 zistíme, že opakovaním tohoto vzoru vieme vytvoriť výhernú stratégiu aj pre každú väčšiu nepárnu šírku. Symetricky to teda vieme aj pre kocku typu J a ešte nám ostávajú kocky S a Z.

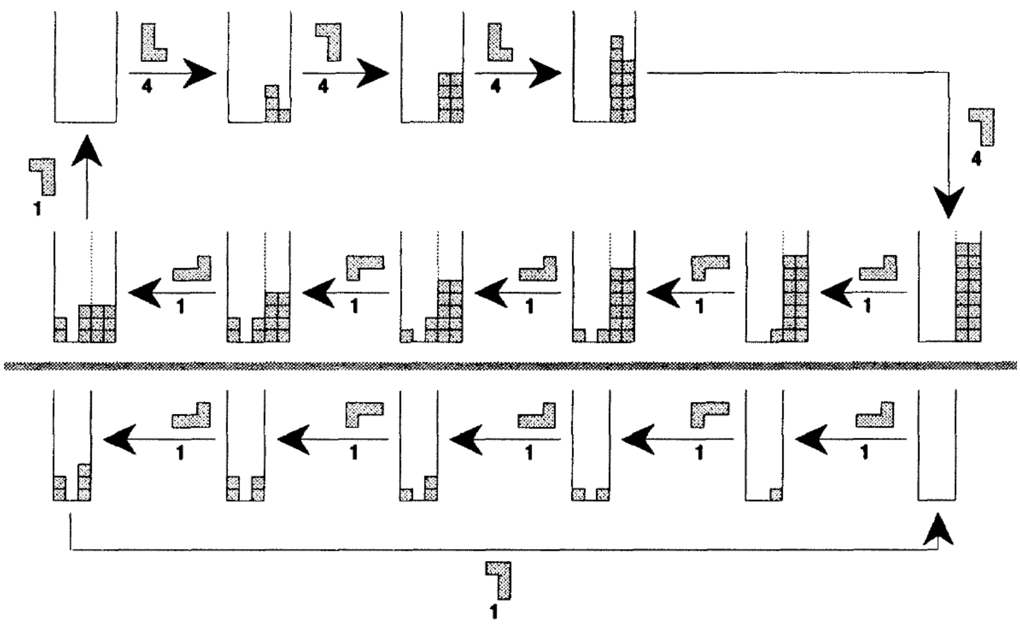
Veta 2.3. *Pre kocky typu S a Z neexistuje výherná stratégie pre Dobrého hráča pri nepárnej šírke hracieho poľa*

Dôkaz. Dôkaz toho, že neexistuje výherná stratégia pre Dobrého hráča je v podstate jednoduchý. Na obrázku 2.9 vidíme, že nezáleží na tom, ako tieto kocky uložíme do hracieho plánu, pri uložení vždy zaplníme rovnaký počet štvorčekov v párnych stĺpcoch („párne štvorčeky - even cells“ - označené **e**) ako v nepárnych stĺpcoch („nepárne štvorčeky - odd cells“ - neoznačené). Budeme sledovať **rozdiel** počtu zaplnených štvorčekov v párnych stĺpcoch a v nepárnych stĺpcoch. Na začiatku hry je tento rozdiel rovný nule. Obrázok ukazuje, že ak uložíme S alebo Z kocku do hracieho plánu a nezmaže sa riadok, tento rozdiel sa nezmení, keďže sa určite pridájú 2 štvorčeky do párnych stĺpcov a taktiež 2 štvorčeky do nepárnych stĺpcov. Keďže hrací plán má nepárnu šírku, zmazaním riadku sa zmaže o jeden štvorček viac v nepárnych stĺpcoch a tento rozdiel sa zvýši. Ak by sme riadky nemazali, výška hry by rástla a zákonite by viedla k prehre Dobrého hráča. Ak budeme riadky mazať, tak sa bude zväčšovať rozdiel, teda bude viac zaplnených štvorčekov v párnych stĺpcoch ako v nepárnych. Keďže máme konečný počet štvorčekov v hracom pláne, toto bude nakoniec tiež viesť k prehre Dobrého hráča. ■

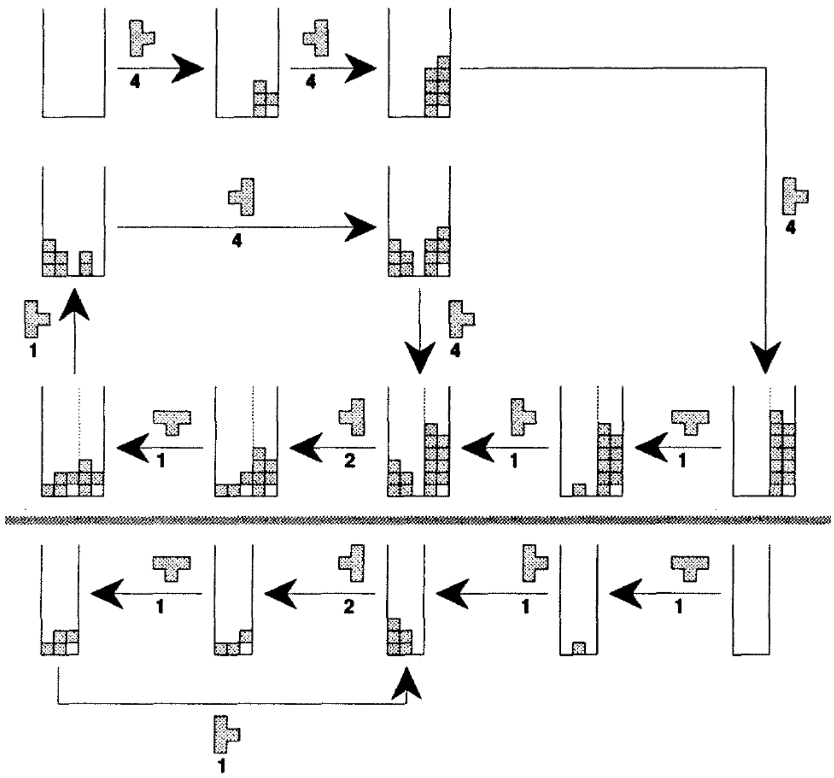
2.3 Využitie základných stratégií

Ukázali sme, že pre každú samostatnú kocku a párnu šírku hracieho poľa existuje výherná stratégia pre Dobrého hráča. Tiež sme ukázali, že pri nepárnej šírke pre niektoré kocky existuje výherná stratégia pre Dobrého hráča a pre ostatné neexistuje (vtedy Dobrý hráč prehrá). Pokúsime sa tieto poznatky využiť k tvorbe Ohodnocovacej funkcie. Našou snahou bude vyrobiť takú Ohodnocovaciu funckiu, ktorá zabezpečí pri simulácii hry rovnaký výsledok ako teória.

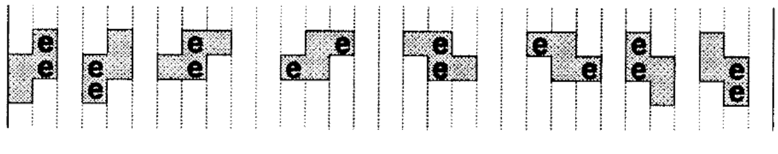
V prípadoch, kde neexistuje výherná stratégia, je výsledok simulácie ľahko zistiteľný. Zlý hráč donúti Dobrého hráča prehrať a tým sa ukončí hra a teda aj simulácia. Pri výherných stratégiách je ale hra nekonečná. To znamená, že aj simulácia hry je nekonečná. Bolo by však vhodné, aby sme aj takúto nekonečnú hru vedeli odhaliť v konečnom čase. Majme hracie pole veľkosti $10 \cdot 20 = 200$ štvorčekov. Každý štvorček može byť v dvoch



Obr. 2.7: Nepárne šírky a kocka L



Obr. 2.8: Nepárne šírky a kocka T



Obr. 2.9: Nepárne šírky a kocky S a Z

stavoch, zaplnený alebo nezaplnený. Pri takejto veľkosti hracieho poľa existuje najviac $2 \cdot 2 \cdot \ldots \cdot 2 = 2^{200}$ rôznych hracích plánov. Toto číslo je nadhodnotené. V skutočnosti je počet rôznych hracích plánov menší, pretože napríklad nemôžeme mať riadky, ktoré majú všetky štvorčeky zaplnené (pretože každý riadok sa v okamihu svojeho zaplnenia vymaže). Podstatné však je, že je toto číslo konečné. To znamená, že ak je hra nekonečná, musia sa nám opakovať niektoré hracie plány (keďže tých je konečný počet).

Definícia 2.5. *Cyklus* je postupnosť hracích plánov, ktorá začína a končí v tom istom hracom pláne.

Na dokázanie nekonečnej simulácie je potrebné v nej nájsť cyklus. Pamätať si a testovať všetky hracie plány od začiatku hry by ale bolo výpočtovo aj pamäťovo náročné, preto musíme zvoliť iný postup. Využijeme poznatky Ohodnocovacej funkcie, ktorá sa snaží jednoznačne ohodnotiť hrací plán. Niekedy sa stane, že dva rôzne hracie plány ohodnotí rovnakou hodnotou. Preto si po každom $25 \cdot 2^i, 0 \leq i, i \in Z;$ kroku zapamätáme postupnosť 10 hodnôt predchádzajúcich hracích plánov a tú každým krokom budeme testovať s novými prichádzajúcimi postupnosťami až do $25 \cdot 2^{i+1}$ kroku. Potom zoberieme novú 10-člennú postupnosť a budeme testovať ďalej. Ak sme týmto algoritmom našli dve rovnaké postupnosti, našli sme aj cyklus a teda aj nekonečnú hru, nekonečnú simuláciu. Výsledok sme si potom pozreli podľa výpisov a skontrolovali správnosť vyhlásenia programu. Ani raz sa nám nestalo, aby sa tento algoritmus pomýlil a chybne označil cyklom niektorú postupnosť kociek. Bolo to hlavne preto, že pravdepodobnosť výskytu dvoch postupností s rovnakými hodnotami Ohodnocovacej funkcie a rôznymi hracími plánmi je príliš malá.

V nasledujúcej časti ukážem niekoľko simulácií. Zlý hráč bude mať veľmi jednoduchú úlohu. V množine typov kociek, ktoré môže použiť bude mať vždy len jednu kocku. V každom ťahu teda Dobrý hráč bude musieť uložiť rovnaký typ kocky. Celá váha preto ostáva na Ohodnocovacej funkcií, ktorá musí správne určiť, ktorý plán je pre Dobrého hráča výhodnejší.

Skušali sme veľké množstvo rôznych funkcií. Pri ich tvorbe sme vychádzali z poznatkov získaných z Parlanteho práce [5]. Premenné sme nastavovali vždy o hodnotu ±0.5. Rádovo sme vyskúšali niečo cez stovku funkcií, ktoré sme väčšinou hneď testovali v zložitejších podmienkach. Dve najzaujímavejšie sú v nasledujúcej časti. Predtým si ale ešte definujeme niekoľko použitých premenných:
MV = (Maximálna) Výška najvyššieho stĺpca v hracom pláne
PV = Priemerná Výška stĺpcov v hracom pláne
PVD = Počet Veľkých Dier v hracom pláne
PSD = Počet Stredných Dier v hracom pláne

PMD = Počet Malých Dier v hracom pláne
HHP = Hodnota Hracieho Plánu
Hĺbka hry je hĺbka výpočtového stromu, počet ťahov ktoré testuje Dobrý hráč predtým ako uloží kocku.

2.3.1 Simulácia 1

Hrací plán: 10 × 20 štvorčekov
Hĺbka hry: 1
Ohodnocovacia funkcia:

$$50 \cdot MV + 40 \cdot PV + 20 \cdot PVD + 15 \cdot PSD + 12.5 \cdot PMD = HHP$$

Simulácie nám dali prijateľné výsledky. Úspešne ukázali cyklickosť hry pre kocky O, I, S a Z, no simulácie kociek L, J a T skončili prehrou Dobrého hráča. Bola to ale prvá funkcia, s ktorou sa podarilo ukázať cyklickosť kociek S a Z. Všetky výstupy programu si môžeme pozrieť na dátach zo simulácie [6].

2.3.2 Simulácia 2

Pri analyzovaní rôznych výsledkov simulácií a ich výstupov možno dôjsť k záveru, že je dôležitá istá vyváženosť medzi výškou hracieho plánu a počtom dier. Z výsledkov simulácií bolo viditeľné, že dieram prináleží väčšia váha, ako by sa mohlo zo začiatku zdať. Značnou úpravou predchádzajúcej funkcie sa nám podarilo vytvoriť celkom prijateľnú Ohodnocovaciu funkciu, ktorej výsledky vidíme v nasledujúcich troch simuláciách:
Hrací plán: 10 × 20 štvorčekov
Hĺbka hry: 1
Ohodnocovacia funkcia:

$$30 \cdot MV + 20 \cdot PV + 30 \cdot PVD + 25 \cdot PSD + 12.5 \cdot PMD = HHP$$

Simulácie s touto Ohodnocovacou funkciou viedli k cyklickosti kociek O, I, S, Z a L. Pre kocky J a T síce viedli k prehre Dobrého hráča, ale hra trvala oveľa dlhšie, čo bol pozitívny výsledok. Výstupy programu je možné vidieť na dátach zo simulácie [7].

2.3.3 Simulácia 3

Predchádzajúca funkcia vyzerala sľubne pre ďalšie testovanie. Problém pre Dobrého hráča spočíval v tom, že hracie pole široké 10 štvorčekov už bolo príliš veľké (hlavne keď sa hráč nepozerá dopredu a nevidí, aké situácie môžu

nastať). Preto sme sa v tejto simulácii rozhodli zvýšiť hĺbku hry na 2. To znamená, že Dobrý hráč sa pozrie ešte o jeden krok dopredu, ako by mohla pokračovať hra a podľa toho sa rozhodne, aký ťah spraví. Nastavenia:
Hrací plán: 10×20 štvorčekov
Hĺbka hry: 2
Ohodnocovacia funkcia:

$$30 \cdot MV + 20 \cdot PV + 30 \cdot PVD + 25 \cdot PSD + 12.5 \cdot PMD = HHP$$

Tieto nastavenia viedli k výsledku, ktorý sme chceli dosiahnuť. Simulácie ukázali cyklickosť všetkých siedmich kociek pri šírke 10. Táto funkcia splnila počiatočné predpoklady, môžeme ju preto považovať za **najuspokojivejšiu** Ohodnocovaciu funkciu, ktorú sme počas nášho hľadania našli. V dátach [8] si môžeme pozrieť výsledok simulácií.

2.3.4 Simulácia 4

V tejto simulácii ešte ukážeme, že predchádzajúca Ohodnocovacia funkcia dáva uspokojivé výsledky aj pre párne šírky menšie ako 10 bez nutnosti zvyšovania hĺbky hry.
Hrací plán: 8×20 štvorčekov
Hĺbka hry: 1
Ohodnocovacia funkcia:

$$30 \cdot MV + 20 \cdot PV + 30 \cdot PVD + 25 \cdot PSD + 12.5 \cdot PMD = HHP$$

Dobrý hráč sa pozeral len na konkrétny ťah, nepozeral sa ako by mohla nasledovať hra v ďalšom ťahu a simulácie aj napriek tomu ukázali cyklickosť všetkých siedmich kociek. To len potvrdzuje predpoklad, že táto funkcia je zatiaľ najlepšou Ohodnocovacou funkciou. Tiež to ale znamená, že čím širšie bude hracie pole, tým sa dobrý hráč bude musieť pozerať do väčšej hĺbky. To ale ešte viac zvyšuje výpočtovú náročnosť, ako z hľadiska hĺbky hry, tak aj z hľadiska šírky hracieho poľa. Výsledky zo simulácií nájdeme v dátach [9].

2.4 Najlepšia Ohodnocovacia funkcia

Z veľkého množstva funkcií sme v predchádzajúcej časti ukázali simulácie dvoch najúspešnejších Ohodnocovacích funkcií. Jedna funkcia nám dodala obzvlášť uspokojivé výsledky. Bola to funkcia:

$$30 \cdot VyskaNajvyssiehoStlpca + 20 \cdot PriemernaVyskaStlpcov + 30 \cdot PocetVelkychDier + 25 \cdot PocetStrednychDier + 12.5 \cdot PocetMalychDier = HodnotaHraciehoPlanu$$

Aby sme ukázali, že je to uspokojivá funkcia podľa našej definície, musíme ešte ukázať jej vlastnosti pri nepárnych šírkach. To znamená, že pre kocky L, J, I a T sme chceli ukázať cyklickosť hry a naopak, pre kocky O, S a Z mala hra viesť k prehre Dobrého hráča. Ako nastavenia simulácií zvolíme:
Hrací plán: 9×20 štvorčekov
Hĺbka hry: 2
Ohodnocovacia funkcia:

$$30 \cdot MV + 20 \cdot PV + 30 \cdot PVD + 25 \cdot PSD + 12.5 \cdot PMD = HHP$$

Tieto simulácie dodali predpokladané výsledky stratégií a tým ukázali, že túto funkciu môžeme považovať za **najuspokojivejšiu** Ohodnocovaciu funkciu. Výsledky si môžeme pozrieť v dátach [10].

Kapitola 3

Zložitejšie stratégie a ich dôsledky

V predchádzajúcej kapitole sme sa venovali základným stratégiám, ktoré opísal vo svojej práci Brzustowski[2] a pomocou ktorých sme vytvorili Ohodnocovaciu funkciu. V tejto kapitole ukážeme niektoré ďalšie Brzustowského stratégie a pokúsime sa pomocou nich ukázať, že nedokážeme vyrobiť **dobrú** Ohodnocovaciu funkciu podľa našej definície.

3.1 Kocka O a k nej kocky I, J alebo L

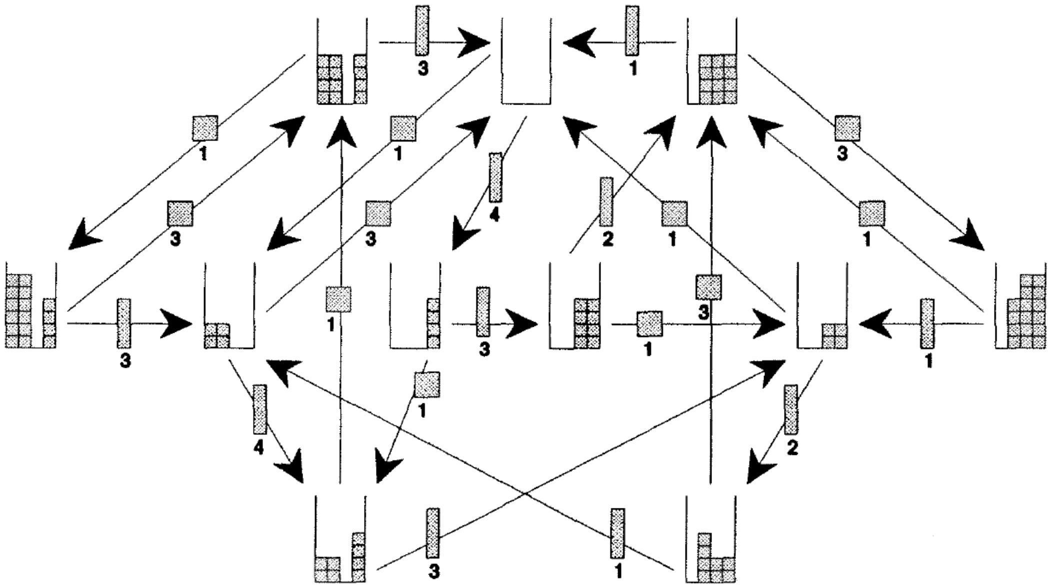
Doteraz sme ukázali výherné stratégie pre hry s jedným typom kocky. Ďalší logický krok je ukázať spôsob, ako hrať tetris s dvomi rôznymi kockami. Na obrázku 3.1 vidieť výhernú stratégie pre kocky typu I a O a šírku hracieho poľa 4 štvorčeky. Podobne na obrázku 3.2 vidieť výhernú stratégiu pre kocky J a O (symetricky to teda platí aj pre kocky L a O). Pokusíme sa teda dokázať, že tieto stratégie platia pre každú párnu šírku hracieho poľa.

Definíciu komína sme uviedli už predtým. Teraz si definujeme ďalšie pojmy.

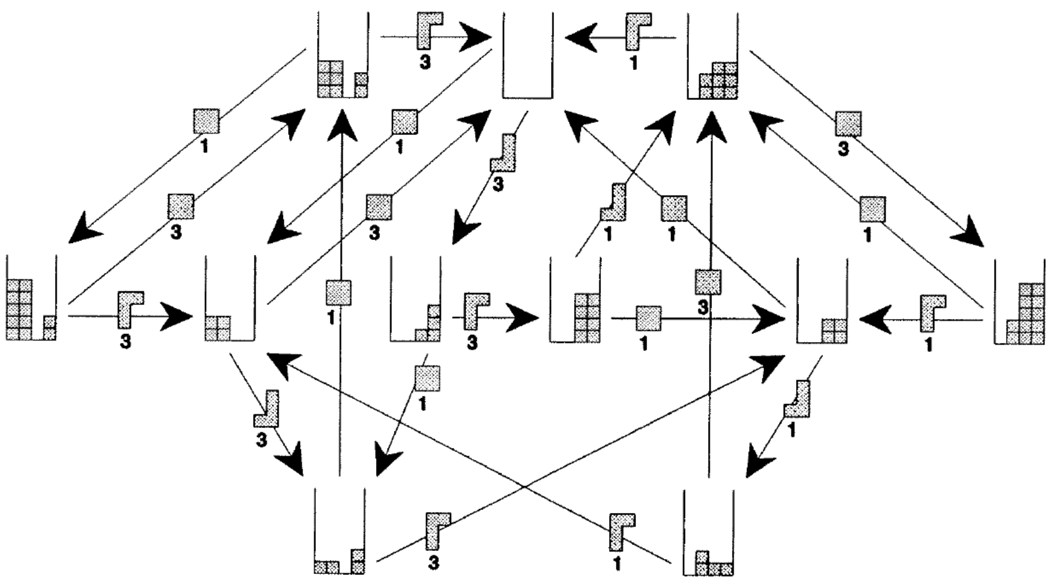
Definícia 3.1. *Výška komína* je výška najvyššieho zaplneného štvorčeka v komíne.

Definícia 3.2. *Zarovnaný komín* je taký komín, ktorý má vo svojej **výške** zaplnené oba štvorčeky v každom riadku (alebo ktorý má nulovú výšku, ak je komín prázdny).

Definícia 3.3. *Hrboľatý komín* má aspoň jeden riadok, v ktorom má zaplnený práve jeden štvorček.

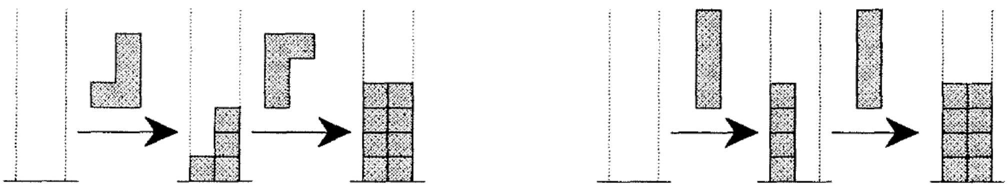


Obr. 3.1: Výherná stratégia pre kocky I a O a šírku 4.

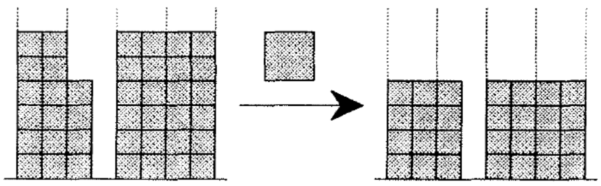


Obr. 3.2: Výherná stratégia pre kocky J a O a šírku 4.

Na predchádzajúcich dvoch obrázkoch vidno, že v stratégii je nanajvýš jeden hrboľatý komín. To je preto, že kocka typu O nevytvára hrboľatý komín a s druhou kockou (kocka I, J alebo L) vieme veľmi ľahko vytvárať a zarovnávať hrboľatý komín (obrázok 3.3).



Obr. 3.3: Ak uložíme kocku I alebo J/L do zarovnaného komína, vytvoríme hrboľ. Ak ju tam vložíme opäť, hrboľatý komín zarovnáme.



Obr. 3.4: Uloženie kocky O do hrboľatého komína zmaže 2 riadky.

Veta 3.1. *Pre dvojice kociek O-I, O-J a O-L existujú výherné stratégie pre Dobrého hráča pri každej párnej šírke hracieho poľa.*

Dôkaz. V tomto prípade Brzustowski definoval nasledujúce pravidlá pre dané kocky, aby dokázal výhernú stratégiu pre každú párnu šírku hracieho poľa:

- **Kocka typu O:** Ak v hracom pláne existuje hrboľatý komín a jeho výška je aspoň o dva štvorčeky menšia ako výška ostatných komínov, ulož kocku O do hrboľatého komína. Inak ju ulož do najnižšieho zarovnaného komína.

- **Iná kocka (kocka I, J alebo L):** Ak v hracom pláne existuje hrboľatý komín, ulož kocku do tohoto komína a zarovnaj hrboľ (vytvor zarovnaný komín). Inak ulož kocku do najnižšieho zarovnaného komína (vytvor hrboľatý komín).

Uvedomme si, že ak sú všetky komíny v hracom pláne zarovnané, aspoň jeden komín je prázdny (má nulovú výšku). Tým pádom ak ukladáme nejakú kocku do najnižšieho zarovnaného komína, ukladáme ju do prázdneho komína. Tiež platí, že v hracom pláne je nanajvýš jeden hrboľatý komín. Vidíme, že jediný problém môže robiť kocka typu O, ak ju ukladáme do hrboľatého komína. Ako ale vidíme na obrázku 3.4, ak ukladáme kocku do hrboľatého komína, ktorý ma výšku aspoň o dva štvorčeky menšiu ako zvyšné komíny, bude táto kocka aj s prislúchajúcimi dvomi riadkami zmazaná a neporuší pravidlo, že je najviac jeden hrboľatý komín. Takto mamé vytvorenú stratégiu pre dvojice kociek O-I, O-J a O-L, ktorá platí pre každú párnu šírku hracieho poľa. ■

3.2 Dvojica kociek I a L, I a J alebo L a J

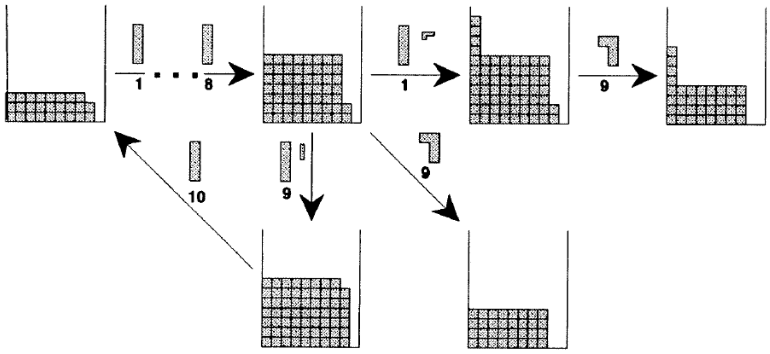
Keď budeme hrať tetris s kockami typu I a L alebo I a J, alebo L a J, výherná stratégia je veľmi podobná ako v predchádzajúcej časti. Podstata je opäť v tvorení čo najmenšieho počtu hrboľatých komínov, ale v tomto prípade budeme potrebovať dva, pre každú kocku jeden. Ak je v hracom poli hrboľatý komín oboch typov kociek, príchodom ďalšej kocky sa jeden z nich stane zarovnaným. Takto vzniká nový problém, ktorý budem, podobne ako Brzustowský, ilustrovať na kocke I a L. Predstavme si, že v určitom prípade je v hracom poli hrboľatý komín, povedzme l, ktorý potrebuje kocku L na to aby mohol byť zarovnaný. Takýto komín budeme volať **L-žiadúci** komín.

Definícia 3.4. *„X“-žiadúci* komín je taký, ktorý potrebuje kocku „X“ k tomu, aby mohol byť zarovnaný.

Čo sa ale stane, ak teraz príde dlhá postupnosť kociek typu I? Ako v predchádzajúcom prípade ich budeme ukladať do iných komínov ako l. Takto nám postupne bude rásť výška týchto komínov. Nakoniec budú všetky tieto komíny zarovnané a ich výška bude najmenej o 4 kocky vyššia ako výška komína l. V tomto bode sa musíme rozhodnúť (pozri obrázok 3.5). Ak máme uložiť kocku L, je postup jasný. Uložíme ju do hrboľatého komína a zrušíme tým 3 riadky a zarovnáme komín. Problém nastane ak príde kocka I. Ak by sme postupovali v ďalšom ukladaní kocky do zarovnaných komínov bez vidiny rušenia riadkov, výška hry by rástla a viedla by k prehre Dobrého hráča. Ak by sme kocku I uložili do hrboľatého komína, zmazali by sme 2 riadky, ale vyrobili by sme J-žiadúci komín, ktorý potrebuje kocku J na to, aby zarovnal komín. Tento problém rieši **náhľad**.

Definícia 3.5. *Náhľad* je možnosť vidieť, aká kocka príde v ďalšom ťahu. Vlastnosť sa štandardne vyskytuje v pôvodnej verzii TETRISu.

Ak vidíme, že v ďalšom ťahu príde opäť kocka I, tak obe kocky uložíme do hrboľatého komína. Tým zrušíme 4 riadky a zase dostaneme L-žiadúci komín. Ak ďalšou kockou bude L, tak kocku I uložíme do zarovnaného komína (aby nám neprekážala) a následne v ďalšom ťahu uložíme kocku L do hrboľatého L-žiadúceho komína. Takto zmažeme 3 riadky a zarovnáme komín.

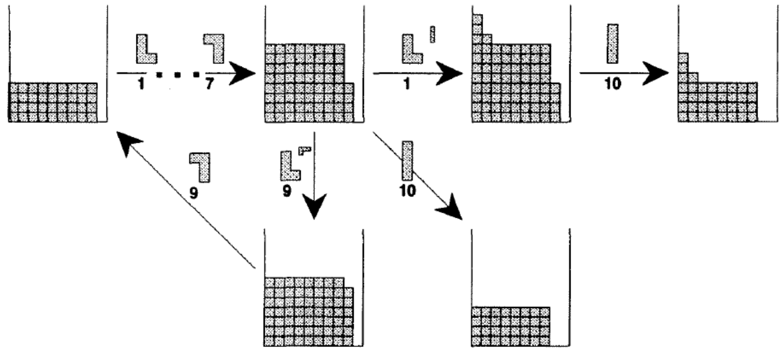


Obr. 3.5: Použitie náhľadu pre kocky I a L. Ak je v hracom poli L-žiadúci komín a príde dlhá postupnosť kociek I, hráč ich ukladá ako vidieť na obrázku, až kým sa hrací plán nedostane do druhého stavu. Tu sa v závislosti od náhľadu kocky v ďalšom ťahu rozhodne, či uloží kocku I do zarovnaného komína, alebo dve kocky I uloží do L-žiadúceho komína. V každom prípade nakoniec skončí v stave s najviac jedným hrboľatým komínom a s nižšou výškou zarovnaných komínov.

Tento postup funguje aj pri vzniku I-žiadúceho komína a dlhej postupnosti kociek typu L (obrázok 3.6), alebo ak vznikne L-žiadúci komín a príde dlhá postupnosť kociek typu J resp. vznikne J-žiadúci komín a príde dlhá postupnosť kociek typu L. Všimnime si ešte, že v hracom poli sa nenachádzajú naraz viac ako dva hrboľaté komíny a počet riadkov v týchto komínoch, ktoré obsahujú jeden štvorček, je najviac 4. Tiež platí, že rozdiel výšok žiadnych dvoch komínov nebude viac ako fixná hodnota. To je postačujúce na dokázanie výhernej stratégie pre dostatočne vysoké hracie pole. Podľa Brzustowského [2] je výška výhernej stratégie s použitím náhľadu kocky o jeden ťah dopredu 10 pre dvojicu kociek L a J a 11 pre dvojicu kociek L a I, alebo J a I.

3.3 Simulácia

Aká úspešná bola naša najuspokojivejšia Ohodnocovacia funkcia pri použití dvoch kociek? V tomto prípade si už Zlý hráč vyberá z dvoch kociek, ktoré môže hodiť Dobrému hráčovi a tak má tiež možnosť zasiahnuť do priebehu



Obr. 3.6: Ak je v hracom poli I-žiadúci komín a príde dlhá postupnosť kociek L, hráč ich ukladá ako vidieť na obrázku, až kým sa hrací plán nedostane do druhého stavu. Tu sa v závislosti od náhľadu kocky v ďalšom ťahu rozhodne, ako kocku uloží. Nakoniec skončí v stave najviac s jedným hrboľatým komínom a s nižšou výškou zarovnaných komínov.

hry. Na simuláciu som zvolil referenčný model:
Hrací plán: 10×20 štvorčekov
Hĺbka hry: 2
Ohodnocovacia funkcia:

$$30 \cdot MV + 20 \cdot PV + 30 \cdot PVD + 25 \cdot PSD + 12.5 \cdot PMD = HHP$$

Ako sa dalo očakávať, pri dvojiciach kociek O-I, O-L a O-J ukázali simulácie cyklickosť hry. Náš model nedáva Dobrému hráčovi možnosť vidieť, aká kocka príde v ďalšom ťahu, a preto sa simulácie dvojíc kociek I-L, I-J a L-J skončili prehrou Dobrého hráča. Ako sme ukázali, Dobrý hráč sa dostal do stavu, keď si musel vybrať z dvoch rovnocenných hracích plánov a oba by mohli pri vhodnom ďalšom vývoji hry viesť k výhre, ale aj k prehre. Zlý hráč mal teda možnosť ovplyvniť vývoj pre neho vhodným smerom a takto poraziť Dobrého hráča. Nakoniec, výsledky si môžeme pozrieť v dátach [11].

3.4 Dobrá Ohodnocovacia funkcia

V predchádzajúcej časti sme ukázali, že na to, aby sme vytvorili výhernú stratégiu pre dvojice kociek L-J, L-I a J-I, potrebujeme vidieť kocku, ktorá príde v ďalšom ťahu. Teraz ale nastáva problém. Spomeňme si na definíciu **dobrého** ohodnotenia. Dobré ohodnotenie je také, ktoré vie pre dvojicu plánov určiť, ktorý plán je lepší. Napríklad ak prvý hrací plán vedie k prehre a druhý k výhre Dobrého hráča, dobré ohodnotenie musí vybrať druhý plán. Alebo ak prvý hrací plán vedie k hre, ktorá skončí až po 1000 ťahoch a druhý plán vedie k hre ktorá skončí do 200 ťahov, dobré ohodnotenie musí vybrať prvý plán. V predchádzajúcej stratégii prišiel moment, kedy sa Dobrý hráč musel rozhodnúť medzi dvomi rôznymi uloženiami kocky a teda medzi dvomi rôznymi hracími plánmi. Aby sa správne rozhodol, musel vidieť, ktorá kocka príde v ďalšom ťahu. Dobré ohodnotenie ale takú možnosť nemá. Musí si vedieť vybrať len z vlastností týchto dvoch hracích plánov. Oba hracie plány vedú za určitých okolností k výhre, ale aj k prehre. Preto neexistuje možnosť, že by nejaké ohodnotenie vedelo ukázať, ktorý hrací plán je lepší. Z toho teda vyplýva, že nemôže existovať dobré ohodnotenie hracích plánov.

Kapitola 4

Koniec hry

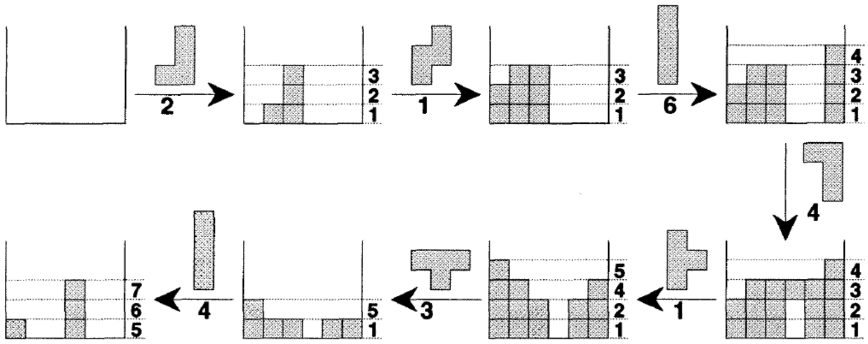
Može byť tetris nekonečný? Môžeme nájsť stratégiu pre Dobrého hráča, aby dokázal ukladať kocky bez prehry? V predchádzajúcej časti sme ukázali, že bez možnosti Dobrého hráča vidieť kocku, ktorá príde v ďalšom ťahu (náhľad), nedokážeme nájsť takú stratégiu a Zlý hráč vie donútiť Dobrého hráča k prehre (stačí mu na to jedna z dvojíc kociek L-I, J-I, alebo L-J). S možnosťou náhľadu ale pre tieto kocky vieme nájsť výhernú stratégiu. V tejto časti sa pokúsime ukázať, že aj s možnosťou náhľadu dokáže donútiť Zlý hráč Dobrého hráča k prehre.

Na to, aby sme mohli ukázať tento Brzustowského[2] dôkaz, musíme ešte predtým vysvetliť niekoľko ďalších pojmov. Jedným z nich je už predtým definovaný pojem cyklu. Pozrieme sa trošku bližšie na jeho život. Predtým si ale ešte definujeme pojem pruh.

Definícia 4.1. *Pruh* je riadok v hracom pláne, ktorý má svoj vlastný život. Pruh môže:

- narodiť sa - akonáhle sa do prázdneho riadku zapíše plný štvorček, vznikne z neho pruh.
- rásť - v pruhu pribúdajú plné štvorčeky.
- klesať - pruh spadne v hracom pláne o jeden riadok nižšie (ak bol nejaký pruh pod ním zmazaný).
- zomrieť - pruh je zmazaný, keď má zaplnené všetky štvorčeky.

Tieto pruhy si začneme číslovať. Na začiatku hry bude počítadlo na nule. Ak sa narodí nový pruh, hodnota počítadla sa zvýši o 1 a pruh dostane toto číslo. Pruh s nižším číslom je starší, ako pruh s vyšším číslom (obrázok 4.1).



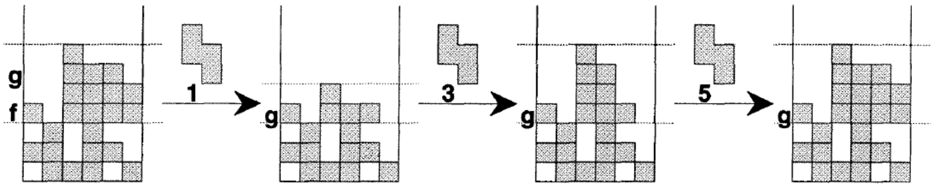
Obr. 4.1: Ukážka života pruhov. Na pravej strane hracieho poľa je napísané číslo pruhu. Poďme si prejsť krok za krokom celú ukážku: (i) pruhy 1, 2 a 3 sa narodili. (ii) pruhy 1, 2 a 3 rastú. (iii) pruh 4 sa narodil a pruhy 1, 2 a 3 rastú. (iv) pruhy 1, 2 a 3 rastú. (v) pruh 5 sa narodil, pruh 3 zomrel a pruh 4 rastie a klesá. (vi) pruhy 2 a 4 zomreli a pruh 5 klesá. (vii) pruhy 6 a 7 sa narodili, pruh 1 zomrel a pruh 5 rastie a klesá.

Povedali sme, že cyklus je postupnosť hracích plánov, ktorá začína a končí rovnakým hracím plánom. To ale neznamená, že sa musí meniť úplne celý plán. Väčšinou sa stane, že sa menia len niektoré riadky a iné ostanú. Vidieť to aj na dátach zo simulácií[8] (hlavne pre kocky S, Z, L, J a T) ale aj na obrázku 4.2.

Definícia 4.2. *Pás cyklu* je množina riadkov, ktoré sa počas cyklu menia.

Brzustowski [2] dokázal niekoľko vlastností pre takýto pás cyklu:

- Ak je v páse cyklu pruh, ktorý počas cyklu neklesol, potom musí zomrieť. Naviac najstarší pruh v páse cyklu zomrie počas cyklu.
- Ak f je pruh v páse nekonečne veľakrát sa opakujúceho cyklu, potom f raz musí zomrieť.
- Ak g je pruh, ktorý sa nachádza v niektorom hracom pláne cyklu a je mladší ako najstarší pruh v páse cyklu, potom sa aj g nachádza v páse cyklu.
- V každom hracom pláne v cykle musia byť pruhy nachádzajúce sa v páse cyklu vždy navrchu hracích plánov (nad všetkými pruhmi, ktoré sa nenachádzajú v páse cyklu).



Obr. 4.2: Cyklus s použitím kocky S. Pás cyklu sú riadky 4 až 7. Vidíme, že riadky 1, 2 a 3 sa v cykle nemenia, preto nepatria do pásu cyklu.

4.1 Porážka pomocou kociek S a Z

V nasledujúcej časti si ukážeme algoritmus, v ktorom dokáže Zlý hráč poraziť Dobrého hráča iba pomocou kociek S a Z. Takto ukážeme, že neexistuje výherná stratégia pre štandardnú verziu hry tetris. Odteraz sa budeme venovať len kockám S a Z.

V každom komíne môže mať pruh nula, jeden alebo dva zaplnené štvorčeky.

Definícia 4.3. *Segment* je oblasť pruhu, ktorú zaberá jeden komín. Symboly □□, □■, ■□ a ■■ reprezentujú 4 možné konfigurácie od prázdnych až po plné štvorčeky v segmente.

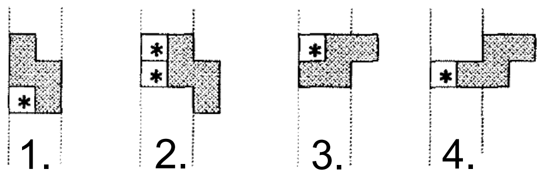
Ak je v pruhu najľavejší segment □■, potom nevieme zabiť pruh kockou S. Tento pruh budeme volať **S-imúnny**. Podobne, pruh s najpravejším segmentom ■□ budeme volať **Z-imúnny**.

Definícia 4.4. Pruh je *„X“-imúnny*, ak sa nedá zabiť používaním len kocky „X“.

Teraz ukážeme, že ak príde dlhá postupnosť kociek S a budeme ju ukladať bez ukončenia hry, vytvorí sa určitý druh štruktúry.

Veta 4.1. *Ak je cyklus tvorený iba kockou S, potom musí byť každá kocka uložená práve do jedného komína (štvorčeky kocky nesmú byť uložené do dvoh komínov). Naviac pre každý komín v každom stave (hracom pláne) cyklu platí, že časť komína, ktorá sa nachádza v páse cyklu je tvorená 0 a viac segmentami* ■■ *a navrchu má jeden segment* ■□.

Dôkaz. Sposôby, ako môžeme uložiť kocku S do komínov vidieť na obrázku 4.3. Problém na obrázku je so štvorčekom ⧆. Ak by bol prázdny, komín by sa stal Z-žiadúcim, ale v cykle máme len kocky S. To znamená



Obr. 4.3: Štyri rôzne spôsoby uloženia kocky S do komínov.

že zaplnenie štvorčeka by sa dalo len kockou, ktorá by zasahovala zase aj do vedľajšieho komína a teda by sme vo vedľajšom komíne vytvorili opäť štvorček ⧆. Takto by sme sa postupne dostali až k najľavejšiemu komínu a vytvorili v ňom segment □■. Tento segment by ale z tohto pruhu spravil S-imúnny, ktorý nemôže byť zabitý kockou S. To by ale znamenalo, že tento pruh nie je v páse cyklu. Ukladanie nových kociek do tohto pruhu znamená, že je v páse cyklu. Dostávame spor s predpokladom, že každý mladší pruh ako pruh, ktorý sa dostal do pásu cyklu, musí byť zabitý. To znamená, že štvorček ⧆ musí byť zaplnený. Na obrázku 4.3 vidíme štyri možné umiestnenia kocky. Prvé umiestnenie je v poriadku. Zaplnenie štvorčeka ⧆ vieme dosiahnuť tak, že už predtým bola do tohto komína umiestnená kocka rovnakým spôsobom (teda tiež nevytŕčala z komína). V ostatných prípadoch to ale znamená, že predtým tam musela byť kocka, ktorá bola tiež umiestnená naraz do dvoch komínov, čím sa opäť dostávame do 2. až 4. spôsobu umiestenia. Takto postupne by sme znova vytvorili S-imúnny pruh a ukázali spor. Teda jediné možné umiestňovanie kocky S v cykle je do samostatných komínov. Na vrchu komína sa tým pádom môže nachádzať iba segment ■□. Pod ním sa v páse cyklu musia nachádzať len segmenty ■■. Ak by tam bol iný segment, tento pruh by nemohol byť počas cyklu zabitý a teda by sa nenachádzal v páse cyklu. Toto tvrdenie platí symetricky aj pre cyklus obsahujúci iba kocku Z. ■

Práve teraz máme dostatočné množstvo poznatkov na to, aby sme si mohli ukázať algoritmus, ktorým Zlý hráč jednoznačne porazí Dobrého hráča.

Veta 4.2. *Nasledujúci algoritmus generuje postupnosť kociek S a Z, ktorá pre párnu šírku hracieho poľa vedie vždy ku konečnej hre.*

1. *Pošli kocku S a ako ďalšiu kocku zobraz kocku S kým nezistíš cyklus.*
2. *Pošli kocku S a ako ďalšiu kocku zobraz kocku Z.*
3. *Pošli kocku Z a ako ďalšiu kocku zobraz kocku Z kým nezistíš cyklus.*
4. *Pošli kocku Z a ako ďalšiu kocku zobraz kocku S.*
5. *Choď na krok 1.*

Dôkaz. Ukázali sme, že ak hra neskončí v krokoch 1 alebo 3, musí určite prejsť cez cyklus. Podľa vety 4.1, ktorú sme si pred chvíľou dokázali, platí, že sa vytvorí určitý druh štruktúry hracieho plánu. Po kroku 2 majú všetky komíny navrchu segment ■□ a po kroku 4 segment □■. Naviac, všetky tieto komíny majú od výšky najnižšieho komína v hracom pláne až po svoj vrch (bez vrchného segmentu) segment ■■, pretože tieto pruhy boli určite v páse cyklu, keďže sú mladšie ako pruh v najnižšom komíne. To ale znamená, že po kroku 2 sa v hracom pláne nachádza Z-imúnny pruh a po kroku 4 S-imúnny pruh. Z-imúnny pruh nemôže byť zabitý kockou Z, ktorá bude prichádzať v kroku 3, a preto sa cyklus vytvorí nad týmto pruhom. To isté platí aj pre S-imúnny pruh po kroku 4 a prichádzajúce kocky typu S v kroku 1. Takto bude postupne rásť výška hry (keďže budú vznikať riadky, ktoré nebudeme môcť zmazať), čo bude mať za následok neodvratný koniec hry. ■

4.2 Časová zložitosť algoritmu

V predchádzajúcej časti sme ukázali algoritmus (veta 4.2) na porazenie Dobrého hráča.

Veta 4.3. *Časová zložitosť algoritmu na porazenie Dobrého hráča je v každom ťahu $O(n \cdot m)$, kde n je šírka a m výška hracieho plánu.*

Dôkaz. Ak si algoritmus pozrieme pozorne, tak zistíme, že jedinou výpočtovo náročnou úlohou pri ťahu Zlého hráča je testovanie cyklickosti hry. Už predtým sme ukázali jeden spôsob, ktorý bol založený na pravdepodobnostnom princípe. Ten ale využíval už vypočítanú hodnotu z Ohodnocovacej funkcie, ktorá má časovú zložitosť $O(n \cdot m)$ (veľkosť hracieho poľa). Ukážeme preto radšej iné testovanie na zistenie cyklickosti hry.

Toto testovanie bude prebiehať počas 1. a 3. kroku predchádzajúceho algoritmu. Na začiatku 1. alebo 3. kroku vždy najskôr vynuluje počítadlo ťahov. Nech c je ľubovoľné prirodzené číslo[1]. Potom v každom $c \cdot 2^i; 0 \leq i, i \in Z$; ťahu si zapamätá hrací plán, ktorý vznikol v tomto ťahu. Zisťovať cyklickosť bude tak, že v každom novom ťahu porovná nový hrací plán so zapamätaným hracím plánom (pravdaže najprv porovná a až potom si prípadne môže zapamätať nový hrací plán). Ak sú rovnaké, našiel cyklus.

Rozdiel medzi dvoma zapamätaniami hracieho plánu sa postupne bude zväčšovať, až bude väčší ako dĺžka hľadaného cyklu. Vtedy si testovanie zapamätá hrací plán, ktorý sa nachádza niekde v postupnosti cyklu. Do ďalšieho zapamätania sa táto postupnosť stihne celá zopakovať, a preto určite bude kontrolovať ešte zapamätaný hrací plán z predchádzajúceho obehu cyklu. Časová zložitosť testovania v každom ťahu je $O(n \cdot m)$, kde n je šírka a m je výška hracieho poľa. Musí totiž prejsť všetky štvorčeky oboch hracích plánov na to, aby skontroloval, či sú rovnaké. Zapamätanie hracieho plánu tiež zaberie čas $O(n \cdot m)$. ■

Veta 4.4. *Horné ohraničenie časovej zložitosti algoritmu Zlého hráča pre celú hru je $O(n \cdot m^2 \cdot 2^{n \cdot m})$, kde n je šírka a m výška hracieho plánu.*

Dôkaz. Ak sa teraz pozrieme na priebeh celej hry, zistíme, že v 1. a 3. kroku algoritmu získame vždy minimálne jeden riadok hracieho poľa, ktorý sa nedá zničiť. V týchto krokoch Zlý hráč robí stále to isté, hádže rovnaké kocky a testuje cyklickosť hry. V každom ťahu mu test trvá čas $O(n \cdot m)$. Vyššie sme ukázali, že je konečný počet rôznych plánov a to maximálne $2^{n \cdot m}$. Preto sa do tohto počtu už určite zopakuje niektorý hrací plán a nájdeme cyklus. Každým nájdením cyklu prejdeme cez 1. alebo 3. krok a tým pridáme jeden nezmazateľný riadok. Keďže hracie pole má m riadkov, po m nájdeniach cyklu bude hra ukončená prehrou Dobrého hráča. Počet ťahov za celú hru je teda $O(m \cdot 2^{n \cdot m})$. Skoro v každom ťahu sa testuje cyklickosť hry (výnimkou sú ťahy v 2. a 4. kroku algoritmu, ale tých je zanedbateľne málo). Horné ohraničenie časovej zložitosti algoritmu Zlého hráča k donúteniu prehry Dobrého hráča je potom $O(n \cdot m^2 \cdot 2^{n \cdot m})$. ■

[1] Menšie číslo c zabezpečí rýchlejšie nájdenie najmenších cyklov, ale zo začiatku spôsobí väčší počet zapamätaní hracieho plánu. Väčšie číslo c zabezpečí rýchlejšie nájdenie dlhších cyklov ale prvé zapamätanie začne po väčsom počte ťahov, teda neskoršie nájdenie kratších cyklov. Výhodnejšie sú preto menšie čísla, pokojne aj $c = 1$.

Kapitola 5

Záver

V práci sme postupne definovali tetris ako hru dvoch hráčov, Zlého hráča (hádzača) a Dobrého hráča (ukladača). Ukázali sme možný algoritmus na simulovanie takejto hry a snažili sme sa nájsť heuristickú funkciu na ohodnotenie hracieho plánu. Ukázali sme, že neexistuje dobré ohodnotenie plánu a najlepšou Ohodnocovacou funkciou ktorú sme našli bola funkcia

$$30 \times VyskaNajvyssiehoStlpca + 20 \times PriemernaVyskaStlpcov + 30 \times PocetVelkychDier + 25 \times PocetStrednychDier + 12.5 \times PocetMalychDier = HodnotaHraciehoPlanu$$

Dokázali sme niekoľko výherných stratégií v rôznych obmenách, či pravidlách hry pre Dobrého aj Zlého hráča. Pri nepárnej šírke hracieho poľa sme dokázali, že Zlý hráč veľmi ľahko donúti Dobrého hráča k prehre. Stačí ak mu bude hádzať stále rovnakú kocku typu O, S alebo Z. Pri párnej šírke sme tiež ukázali algoritmus, ktorým si Zlý hráč vynúti koniec hry a to s časovou zložitosťou $O(n \cdot m)$ na ťah a celkovým horným ohraničením časovej zložitosti $O(n \cdot m^2 \cdot 2^{n \cdot m})$ na celú hru, kde n je šírka a m výška hracieho poľa. Ak by mal hráč (prípadne počítač), ktorý hádže kocky, tieto poznatky, aj ten najlepší hráč, ktorý ukladá kocky by bol donútený k prehre a **hru TETRIS by nemohol hrať donekonečna**.

Dodatok

K práci je priložené CD so zdrojovými kódmi v jazyku Java a spustiteľným appletom. Taktiež sú súčasťou CD aj súbory s výsledkami niektorých simulácií.

Na stránke `http://fks.sk/~mato/tetris/` sa nachádza spustiteľný java applet s programom a možnosťou vytvárania vlastných jednoduchých Ohodnocovacích funkcií.

Literatúra

[1] http://cs.wikipedia.org/wiki/Pentomino, Hra Pentomino.

[2] J. Brzustowski, "Can you win at TETRIS?," 1988.

[3] http://en.wikipedia.org/wiki/Tetris, Hra TETRIS.

[4] E. Demaine, S. Hohenberger, and D. Liben-Nowell, "Tetris is hard, even to approximate," *Lecture notes in computer science*, pp. 351–363, 2003.

[5] N. Parlante, "Nifty assignments: tetris on the brain," *SIGCSE BULLETIN*, vol. 33, no. 4, pp. 25–27, 2001. http://cslibrary.stanford.edu/112/, Stanford Tetris Project.

[6] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia1/*".

[7] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia2/*".

[8] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia3/*".

[9] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia4/*".

[10] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia5/*".

[11] Dáta zo simulácie programu. Nachádzajú sa v prílohe v adresári "*simulacie/simulacia6/*".