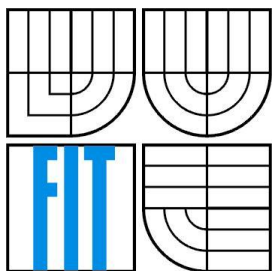


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

REVERSI
REVERSI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ LABAJ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN

BRNO 2007

Zadání bakalářské práce

Řešitel: **Labaj Tomáš**
Obor: Informační technologie
Téma: **Reversi**
Kategorie: Umělá inteligence

Pokyny:

1. Prostudovat pravidla deskové hry Reversi
2. Prostudovat techniky používané pro programování her
3. Navrhnout program pro hraní hry Reversi (člověk vs. PC, PC vs. PC)
4. Uvedený program implementovat

Literatura:

- Internet

Při obhajobě semestrální části projektu je požadováno:

- První tři body zadání

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese
<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Rozman Jaroslav, Ing.**, UITS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Tomáš Labaj**
Id studenta: 84272
Bytem: Ladova 9/290, 736 01 Havířov
Narozen: 30. 04. 1985, Havířov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Reversi
Vedoucí/školitel VŠKP: Rozman Jaroslav, Ing.
Ústav: Ústav inteligentních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

Kabaj
.....

Autor

Abstrakt

Tato práce se věnuje umělé inteligenci, resp. použití umělé inteligence na deskové hře Reversi. Ta je nejlépe realizovatelná pomocí metody minimax. Aby nedocházelo k zbytečnému prohledávání stavového prostoru, je vhodné zavést omezení v podobě prořezávání Alfa-beta. Obě metody jsou zde popsány a vysvětleny. Další část je věnována strojovému učení, tedy tomu, jak může počítač vylepšit svůj tah, když byl minule neúspěšný.

Klíčová slova

Umělá inteligence, UI, Reversi, Othello, hry, Minimax, Alfa-beta prořezávání, strojové učení, ohodnocení, C++, .NET

Abstract

In this bachelor thesis present the problematic of an Artificial Intelligence and its usage for the board game Reversi is described. The best solution for this type of application is “minimax” method. To avoid redundant seeking through status field it is better to use some kind of limitations, e. c. Alpha-Beta method. Both methods are also described in this thesis. Second part is focused on self- learning computer algorithms (the ways how computer can improve his turn after unsuccessful one).

Keywords

Artificial Intelligence, AI, Reversi, Othello, games, MiniMax, Alpha-beta pruning, machine learning, valuation, C++, .NET

Citace

Tomáš Labaj: Reversi, bakalářská práce, Brno, FIT VUT v Brně, 2007

Reversi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaroslava Rozmana.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Labaj
25. dubna 2007

Poděkování

Děkuji ing. Jaroslavu Rozmanovi za podnětné připomínky k programové i textové části bakalářské práce.

© Tomáš Labaj, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	2
2 Umělá inteligence	3
2.1 Definice	4
2.2 Historie	5
3 Hry	6
3.1 Strategické	6
3.2 Stolní, deskové.....	7
3.3 Akční	7
3.4 Konverzační.....	8
3.5 Ostatní.....	8
4 Reversi	9
4.1 Historie	9
4.2 Pravidla.....	9
4.2.1 Základní postavení.....	9
4.2.2 Tah.....	10
4.2.3 Vítězství.....	11
4.3 Strategie	11
4.3.1 Rohy.....	11
4.3.2 Okraj	12
4.3.3 Pohyblivost	12
4.3.4 Rovnováha	13
4.3.5 Předvídavost	13
4.3.6 Zakončení	13
5 Umělá inteligence a Reversi	14
5.1 Stavový prostor.....	14
5.2 Metoda minimax.....	15
5.3 Prořezávání Alfa-beta	17
5.4 Hodnotící funkce	20
5.5 Strojové učení	26
5.6 Ověření inteligence.....	28
6 Závěr	29
Literatura	30

1 Úvod

Tato práce se zabývá umělou inteligencí ve hrách a do větších detailů rozebírá jednu konkrétní deskovou hru jménem Reversi neboli Othello.

Publikace se dá rozdělit na dvě části: teoretickou a praktickou. Teoretická část je obsažena v kapitolách 2 až 4. V páté kapitole, která je jednoznačně největší, je rozebrána praktická část.

Teoretická část

Kapitola dvě definuje a popisuje umělou inteligenci. Také je v ní obsažena podkapitola o historii tohoto vědního oboru.

Jelikož se tato bakalářská práce zabývá hrou Reversi, tak si vymezíme pojmy: umělá inteligence a hry. V další kapitole, tedy třetí, se zaměřuji právě na zmiňované hry. Jaké typy her s umělou inteligencí jsou nejoblíbenější a jaká umělá inteligence je v nich obsažena a co zajišťuje.

Čtvrtá kapitola se již věnuje hře Reversi. Nejprve se v ní píše historie této deskové hry dvou hráčů, poté jsou vysvětleny pravidla, aby čtenář mohl porozumět strategii, která je následující podkapitolou.

Praktická část

Tato část obsahuje pouze jednu kapitolu a to pátou. Tato kapitola se zabývá algoritmy, jakými je umělá inteligence ve hře *Reversi* naprogramována a dala by se rozdělit do dvou podčástí: zjišťování tahů dopředu (metodou minimax) a učení počítače (strojové učení).

V poslední kapitole je popsáno zhodnocení dosažených výsledků ohledně aplikované umělé inteligence. Její potenciální další rozvoj a úpravy.

2 Umělá inteligence

Umělá inteligence je vědní obor, jehož hranice není lehké definovat. Spolupracuje totiž s mnohými dalšími obory, jako jsou např. matematická logika, teorie her, lingvistika atd. Do podoborů umělé inteligence bychom mohli zařadit: počítačové vidění, expertní systémy, genetické programování, genetické algoritmy, reprezentaci znalostí, strojové učení, strojové plánování, neuronové sítě, zpracování přirozeného jazyka, robotiku, umělý život, umělé bytí a distribuovanou umělou inteligenci.

Ale co to vlastně je „umělá inteligence“? Přesná definice tohoto výrazu neexistuje. Je však několik známých definic, které se o to snaží. Některé z těchto definic jsou uvedeny v podkapitole 2.1. Dříve než se k nim však dostanu, bych mohl říct, že umělá inteligence se rozděluje na dva hlavní proudy. Na slabou umělou inteligenci a silnou umělou inteligenci. Slabá umělá inteligence je taková, která splní požadavky Turingova testu, ale neprojde argumentem čínského pokoje. Silná umělá inteligence je schopna i tohoto.

Turingův test

Test spočívá v tom, že do jedné místnosti se umístí počítač a do dalších dvou místností dva lidé. Jeden z nich pokládá otázky a podle náhody odpovídá buď počítač nebo člověk. Tazatel jen ví, z jaké místnosti přichází odpověď na jeho otázku, ale neví, jestli je v ní člověk nebo stroj. Po sérii otázek má říci, kde si myslí, že se kdo (co) nachází.

Argument čínského pokoje

To, že stroj dokáže smysluplně odpovídat na položené otázky ještě neznamená, že rozumí významu otázek a dokonce i jeho odpovědí. Tímto se zabývá argument čínského pokoje, který vymyslel americký filosof J. Searle roku 1980.

Argument čínského pokoje není nikdy možné vyzkoušet na člověku, ale jeho princip je často aplikovaný. Spočívá totiž v tom, že umístíme člověka, který čínštinu neumí, do místnosti, ve které je napsaná každá smysluplná věta v tomto jazyce a zapsaná v čínských znacích. Tento člověk však bude vědět, kde má co hledat, pokud se ho někdo na něco zeptá. Poté mu někdo bude klást otázky (písemně). Člověk, který je uzavřen v této ohromné knihovně, začne hledat tuto otázku a odpověď na ni. Až ji najde, tak symboly pouze přepíše, aniž by rozuměl dané otázce a odpovědi a lístek odevzdá jako odpověď. Nyní by se mohl tazatel domnívat, že odpovídající umí čínsky, ale není tomu tak. Pouze porovnával a opisoval symboly. A tuto práci by mohl zastat i stroj.

2.1 Definice

Minského definice

„Umělá inteligence je věda o vytváření strojů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence“ (Minsky, 1967).

Tato definice nám tedy říká, že řešené úkoly nejsou zcela triviální, ale jsou složitější, jelikož u primitivních úkolů bychom ani u člověka nemohli mluvit o jeho inteligenci. Inteligence je tedy vhodné vybrání té možnosti ze všech, abychom se, pokud možno, co nejdříve a nejefektivněji dostali k požadovanému cíli.

Definice Richové

„Umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zatím zvládají lidé lépe“ (Rich, Knight, 1991).

Tato definice podle mého názoru rozhodně není přesná, ač se snaží o nezávislost času a řešení daných problematik. Nemohli bychom totiž označit za inteligentní takový stroj nebo program, který by měl třeba jen o trochu horší výsledky než člověk. A i takovýto neúspěch by se podle mého takto označit mohl, jelikož by vykazoval známky jisté inteligence. Nemluvě o tom, že tato definice nezahrnuje takové snažení, jaké je řešení úloh, které nezvládají ani lidé.

Kotkova definice

„Umělá inteligence je vlastnost člověkem uměle vytvořených systémů vyznačujících se schopností rozpoznávat předměty, jevy a situace, analyzovat vztahy mezi nimi a tak vytvářet vnitřní modely světa, ve kterých tyto systémy existují, a na tomto základě pak přijímat rozhodnutí, za pomoci schopností předvídat důsledky těchto rozhodnutí a objevovat nové zákonitosti mezi různými modely nebo jejich skupinami“ (Kotek a kol., 1983).

Na základě vstupních informací si vytvoříme počáteční stav. Také musíme vědět jaký stav je cílový (jak vypadá nebo jeho podmínky). Pomocí možných akcí na počátečním modelu se snažíme, aby tyto dva modely byly stejné. Tímto postupem tedy můžeme získat i nové zákonitosti i v reálném světě. Je ovšem třeba, aby byly jasně definované veškeré potřebné vztahy a pravidla.

2.2 Historie

Již jsme si tedy přiblížili, co se asi skrývá pod slovy *umělá inteligence* a tak bychom mohli odhalit i roušku minulosti tohoto oboru.

Uvádí se, že vznik vědní disciplíny „Umělá inteligence“ by se mohl datovat do roku 1956, na malé konferenci v New Hampshire známé jako dartmouthská konference. Zde také vědci pojmenovali tento vědní obor. Avšak již o 6 let dříve známí vědci A. Turing a J. von Neumann se dali slyšet, a vnukli vědcům myšlenku umělé inteligence.

Z dnešního hlediska je také velmi zajímavé číst si předpovědi ve vývoji, které vědci na dartmouthské konferenci přijali. Bylo uvedeno, že za 14 let od této konference – tedy roku 1970 – stroje budou rozumět lidské řeči, budou skládat hudbu, odhalí nové fyzikální vzorce a stroj bude velmistrem v šachu. Dva roky poté byl vyvinut perceptron F. Rosenblattem. Perceptron simuluje nervovou buňku. Tímto se naskytla možnost strojového učení. Ale žádná z dartmouthských předpovědí se nestala a umělá inteligence byla v úpadku. A aby to nebylo málo, dva uznávaní vědci M. Miský a S. Papert naspali o umělé inteligenci knihu, která ji srazila na kolena. Až o několik let později bylo díky „inteligentnímu“ programu odhaleno ložisko molybdenových rud. Po dalších několika letech nastává velký zvrat v odvětví neuronových sítí. Tato technologie byla stále více a více populárnější. V 80. letech se začínají komercializovat tzv. expertní systémy. Tyto systémy simulují rozhodování expertů na danou problematiku. Takže dochází k automatizaci, potažmo k úsporám. Od té doby je nahlíženo na umělou inteligenci stále s respektem.

S tím, jak rychle se vyvíjí technologie počítačů, jsou kladeny větší nároky na umělou inteligenci ve všech jejích částech a oborech. Její vývoj je sice pomalejší než se v roce 1956 čekalo, ale jisté pokroky zaznamenány jsou. Počítače jsou schopny rozumět lidské řeči, skládat hudbu (ač ne na takové úrovni, jak se čekalo), počítač porazil velmistra šachu G. Kasparova a od roku 2005 existuje nejnovější model ASIMO – což je humanoidní robot, který dokáže samostatně chodit, rozpoznává řeč, obličej, gesta a reaguje na ně. Sice k tomuto pokroku došlo se zpožděním 30 let a není tak velký jak se čekalo, ale i přesto je umělá inteligence jeden z nejrychleji se vyvíjejících vědních oborů.

3 Hry

V této kapitole se budeme zabývat hrami. Jaké typy her jsou nejčastější a jaká umělá inteligence je v nich implementována, co všechno řídí a co musí brát v úvahu.

Počítačové hry se stále těší velké oblibě a to samozřejmě vede k jejich vývoji. Především se vyvíjí jejich grafika a umělá inteligence. Ta ještě před 10 lety na tom nebyla v počítačových hrách nejlépe. Program vždy dělal to stejné a náhoda byla potlačena. Takže stačilo, aby si hráč zapamatoval určitý postup a byl neporazitelný. To už teď ve většině příkladů naštěstí neplatí.

3.1 Strategické

Když se řekne: „Umělá inteligence ve hrách“, tak většině lidí vytanou na mysl hry strategické. Tyto hry by se bez umělé inteligence obešly jen velmi těžko a to jen tak, že by hráli proti sobě jen lidé. A to by vždy nebylo dost dobře možné.

Strategické hry mají většinou podobný charakter. Hráč dostane město, vojenskou základnu, kolonii, nebo třeba jen pár pracovníků a má za úkol ovládnout hrací mapu a to tak, že zničí všechny protihráče.

Herní systém má veškeré prvky provázané. Hráč se musí starat, aby mu jeho lidé nevyhladovali, jeho budovy nespadly, měl dostatečnou vojenskou sílu alespoň na obranu. Umělá inteligence je tady přítomna ve dvou částech.

Budování

Umělá inteligence by se měla v této části starat o to, kam, kdy a kolik surovin půjde do daných částí odvětví (stavitelství, vojenství, ...). Taky by měla zajišťovat vhodné rozložení budov.

Zde implementace umělé inteligence není příliš složitá. Pouze se vyrovnávají veškeré složky a nadbytek se může investovat do ofenzívy. Ani rozložení kolonie nebo města není velký inteligenční problém.

Taktika pro vítězství

Počítač musí využít slabé stránky protihráče. Pokud je např. na pokraji hladomoru, je pro něj výhodné ničit pole s obilím a chvíli čekat až soupeř vyhladoví.

V boji by měla umělá inteligence řídit taktiku boje. Odkud, kdy a kolik jednotek zaútočí. V této části hry je umělá inteligence velmi důležitá. Musí být dostatečně výkonná, aby mohla hráče porazit a musí být naprogramována pomocí rychlých algoritmů, jelikož v boji může být jednotek mnoho.

3.2 Stolní, deskové

Deskové hry byly průkopníkem umělé inteligence aplikované do her. Důvod je jednoduchý: mají konečný počet stavů, jelikož hrací deska je většinou omezena a hráči jsou většinou dva. Pravidla jsou jednoduše naprogramovatelná. Vidina toho, že počítač bude hrát s velmistrem v šachách a porazí ho, byla lákavá. A skutečně se to povedlo, jen o mnoho let později než se zpočátku předpokládalo. V roce 1997 se lidský velmistr Garry Kasparov utkal s počítačem od firmy IBM s názvem Deep Blue. Jednalo se o klasickou turnajovou hru – tedy 6 her s hodinami. Deep Blue vyhrál poměrem 3,5 – 2,5. Kasparov několik dní poté požádal firmu IBM o novou hru, protože Deep Blue věděl o všech partiích a taktice všech hráčů, tedy i Kasparova, zatímco Kasparov o Deep Blue nevěděl vůbec nic a netušil, jakou taktiku stroj zaujme. IBM však tento stroj již rozebrala a nehodlala jej znovu skládat. Velmistr se však vyjádřil, že Deep Blue udělal i chyby. Řídil se záznamy jiných her a proto udělal i „lidskou“ chybu.

Jako metoda pro umělou inteligenci v této sekci her se standardně používá algoritmus minimax. Jedná se o prohledávání všech možných stavů a podle toho se volí daný tah. K zlepšení efektivnosti minimaxu se používá metoda alfa-beta.

Deskové hry také často využívají strojového učení, jelikož záznam tahů není časově ani paměťově náročný. Tento faktor je poměrně zajímavý, jelikož pomocí jiných partií se počítač dokáže rozhodnout, kam táhnout.

Do této kategorie spadá i hra Reversi. Takže se s metodami minimax a alfa-beta detailněji seznámíme v kapitole 5.

3.3 Akční

V akčním typu her obvykle hráč ovládá jen jednu postavu z pohledu first person, tedy z pohledu jak by tento hráč svými očima viděl situaci, kdyby stál na místě dané počítačové postavy. Má za úkol někam dojít, zabít nepřítele nebo splnit nějaké úkoly.

Umělá inteligence v těchto hrách ovládá jeho protivníky a v některých hrách i jeho počítačové pomocníky. Obvykle se takové postavě, která je ovládaná počítačem, říká BOT. V prvních hrách tito BOTi byli naprogramováni tak, že chodili od určitého místa k určitému místu a když na své trase zpozorovali protivníka, tak po něm začali střílet. Nyní už se BOTi dokážou pohybovat samostatně po celé hrací mapě a volí si i útočnou taktiku (obklíčení, krytí, ...).

3.4 Konverzační

Konverzační hry mají za úkol srozumitelně mluvit s člověkem. Člověk by měl mít pocit, že mluví s druhým člověkem, který na něj reaguje.

Umělá inteligence zde bývá řešena tak, že z věty si počítač uloží klíčová slova a na ty potom reaguje ve své odpovědi. Na tuto odpověď by měl však dostat nějaký vjem, jestli odpověděl dobře nebo ne – aby se učil. V této oblasti jsou nejznámější dva projekty: ELIZA a A.L.I.C.E. (Artificial Intelligence Internet Computer Entity).

ELIZA je prvním významným projektem. Vyvíjel se několik let, takže má mnoho verzí. V pozdějších verzích byl tento program schopen se i učit novým klíčovými slovy a odvozovat pravidla pro jejich užívání.

A.L.I.C.E. je v dnešní době asi nejlepším programem v této sféře umělé inteligence. Její zakladatel Dr. Richard S. Wallace v roce 2002 dostal Loebnerovu cenu za nejúspěšnější výsledek v Turin-gově testu (tento test je popsán v kapitole 2). A.L.I.C.E. je naprogramována v Javě a v C/C++ a v tuto chvíli pracuje na jejím vývoji více než 300 lidí. Tento projekt si každý může otestovat na stránce <http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1>.

3.5 Ostatní

V ostatních hrách jako např. dobrodružné, arkády, postřehové... umělá inteligence buďto není přítomna vůbec nebo je zanedbatelná.

4 Reversi

Reversi neboli Othello (oba tyto názvy jsou platné a používají se) je desková hra pro dva hráče, která je rozšířena po celém světě.

4.1 Historie

Hra *Reversi* byla poprvé popsána roku 1880 v Anglii pány L. Watermanem a J. W. Mollettem. O několik let po jejím uveřejnění se stala velkým hitem. Avšak roku 1898 v Německu jeden časopis zabývající se hrami otiskl článek, ve kterém píše o nové hře, kterou vynalezl, a pravidla jsou téměř totožná s pravidly anglickými. Roku 1970 byla v Japonsku uznána nová světově jednotící pravidla této hry pod jménem Othello. Toto jméno nebylo vybráno náhodou, ale proto, že divadelní postava napsána perem W. Shakespeara o sobě tvrdí, že je muž dvou tváří, a hrací kameny Othella mají také dvě strany různých barev.

Od roku 1977 se hraje World Othello Championship – tedy Mistrovství světa v Othellu. Na tomto mistrovství dominuje Japonsko, které v této hře vyniká. Samozřejmě, že na tomto mistrovství hrají jen lidé a ne počítače. Člověk, na rozdíl od hry šachy, totiž proti superpočítači v této hře nemá šanci. Již v roce 1980 byl stávající mistr světa poražen počítačem. A v roce 1997 další počítačový program dosáhl proti tehdejšímu mistru světa výsledku 6:0.

4.2 Pravidla

4.2.1 Základní postavení

Na obrázku Obr. 4.1 vidět základní postavení hry Reversi. Kdysi byla herní deska úplně prázdná a první dva tahy každého hráče musely směřovat od středového čtverce, ale od tohoto pravidla se upustilo před několika lety. Herní deska je standardně velká 8x8 polí a její oficiální název je *othellier*. Další rozměry, na kterých se obvykle také hraje jsou 6x6 a 10x10 polí.

	A	B	C	D	E	F	G	H
1								
2								
3								
4				●	○			
5				○	●			
6								
7								
8								

Obr. 4.1 Základní postavení hry

4.2.2 Tah

Hráč, který je na tahu, umístí svůj kámen tak, aby se mezi tímto kamenem a ještě jedním jeho kamenem (ve všech osmi směrech) nacházela alespoň jedna soupeřova figura. Ta změní po tomto tahu svou barvu – tedy reversuje barvu hráče. Poté je na řadě soupeř. Příklad je patrný na obrázku Obr. 4.2, kde po tahu hráče s bílými kameny na C4 umístil jeho soupeř svůj kámen na B4 a tím získal i tři bílé kameny. Na Obr. 4.3 může čtenář vidět situaci, kdy hráč s bílými kameny umístil kámen tak, že obrátí soupeřovy kameny ve dvou směrech – horizontálně a diagonálně.

Pokud hráč, který je na tahu, nemůže svůj kámen umístit na žádné takové políčko ve hře, vzdává se tahu a pokračuje jeho protihráč. Pokud ani tento hráč nemůže nikam položit kámen se svou barvou, hra končí. Všechny pole na hrací desce tedy nemusí být nutně obsazeny, aby hra skončila.

	A	B	C	D	E	F	G	H
1								
2								
3					○			
4			○	○	○	●		
5				○	●			
6				●	○			
7								
8								

	A	B	C	D	E	F	G	H
1								
2								
3					○			
4		●	●	●	●	●		
5				○	●			
6				●	○			
7								
8								

Obr. 4.2 Tah hráče s černými kameny

	A	B	C	D	E	F	G	H
1								
2						●		
3				○	●			
4			●	●	●	●		
5				○	○			
6					○			
7								
8								

	A	B	C	D	E	F	G	H
1								
2						●		
3				○	○	○		
4			●	●	○	●		
5				○	○			
6					○			
7								
8								

Obr. 4.3 Tah hráče s bílými kameny

4.2.3 Vítězství

Vítězí ten hráč, který má na konci hry více svých kamenů. Také existuje hra nazvaná AntiReversi. Tato hra má úplně stejná pravidla jako hra Reversi, jen ukončení má opačný charakter. U AntiReversi vyhrává ten hráč, jenž má kamenů své barvy méně.

4.3 Strategie

Začátečníci, kteří hráli tuto hru jen opravdu málokdy, většinou hrají tak, aby jejich momentální tah obrátil pokud možno největší počet soupeřových kamenů. Je to však většinou jen dočasné. Soupeř si lehce může připravit takovou situaci, že potom v několika málo tazích zvrátí celou hru na svou stranu a vyhraje.

Je však několik rad, kterými by se měl hráč řídit, jestli chce vyhrát. Jsou to:

4.3.1 Rohy

Rohy hrací desky jsou jedny ze stěžejních bodů hry. Proč? Důvod je jednoduchý: pokud jeden z hráčů obsadí libovolný z rohů, už mu tento bod určitě zůstane do konce hry. Protihráč již nemá jakýkoliv způsob mu tento kámen obrátit – tedy změnit jeho barvu. To však není jediná výhoda rohu. Pomocí něj můžeme obrátit velký počet soupeřových kamenů. Takto získané hrací figury na diagonální straně sice můžeme rychle ztratit, ale u horizontálních a vertikálních můžeme mít velkou pravděpodobnost, že nám již zůstanou. Takže oba hráči se snaží obsadit nejvíce rohů, jak jen to je možné. Ale aby to nebylo tak jednoduché – umístění kamene na rohové místo hrací desky příliš brzy není velká výhoda. Spíše se jen jedná o eliminaci možné výhody soupeře. Brzy získaný roh nám totiž nemusí zaručit ani jeden další bod.

4.3.2 Okraj

Okraj má podobné taktické vlastnosti jako roh, ale není až tak důležitý. Může se totiž snadno stát, že obsadíme skoro celý okraj jedné strany, ale pokud nemáme roh, tak celou tuto řadu můžeme ztratit v jednom tahu. Proto se nepokoušíme bezhlavě získat každé pole na okraji a pokud již nějaké pole na něm máme obsazené, snažíme se tuto stranu udržet.

Nejlepší je držet se pokud možno ve středu hrací desky. Snažíme se nezpřístupnit kraj či roh soupeři tím, že umístíme náš kámen před tyto řady – tedy do kritické oblasti (viz. Obr. 4.4).

	A	B	C	D	E	F	G	H
1								
2		■	■	■	■	■	■	
3		■					■	
4		■		●	○		■	
5		■		○	●		■	
6		■					■	
7		■	■	■	■	■	■	
8								

Obr. 4.4 Kritická oblast

4.3.3 Pohyblivost

Tento typ strategie by se také mohl jmenovat: početnost tahů. Snažíme se eliminovat počet soupeřových tahů, zvláště takových, které by nám mohly „ublížit“. Pokoušíme se tedy táhnout tak, aby soupeř neměl moc polí na výběr kam táhnout a my už před jeho tahem přibližně věděli, kam potáhne. Samozřejmě, že nejlépe pro nás je udělat takový tah, který nám zaručuje, že soupeř nebude mít žádné možné táhnoucí pole, takže bychom táhli opět my.

Na druhou stranu, musíme si dávat pozor, abychom se nedostali do situace, kde bychom neměli možnost nikam umístit náš kámen. Ač v jednom tahu máme možnost táhnout např. na 10 míst, tak soupeřův tah může toto způsobit. Tato chyba by mohla dát soupeři velikou výhodu. Dokonce by mohl začít uvažovat jak pokračovat dál s možností naší tahové eliminace.

Navzdory všemu řečenému k této problematice, existují i expertní taktiky, které se mimo jiné zaměřují na pohyblivost. Hráč si schválně nechává pouze několik málo svých hracích figur a kontroluje si taktické body na hrací desce. Až přijde výhodná chvíle, zaútočí a jeho řady figur nebudou žádnými kameny rušeny, takže jeho útok může být zdrcující. Avšak tuto taktiku ovládá jen othellovská špička a pro začátečníky nebo pokročilé se příliš nedoporučuje.

4.3.4 Rovnováha

Jak hra plyne, tak se na hrací desce vytváří několik oblastí, ve kterých může mít jeden z hráčů přesilu. Snadno si můžeme spočítat, kolik tahů můžeme v oblasti, kterou ovládáme udělat, aniž bychom dali nějakou výhodu soupeři. Takto si můžeme spočítat i soupeřovy tahy v jeho oblasti. A dozvíme se, kdo a za jak dlouho se dostane pod tlak a musí udělat nějakou chybu. Pokud jsme to my, tak si snažíme v našich tazích udělat nějakou výhodnější pozici. Velká chyba je v této chvíli obsazení předrohového kamene (Obr. 4.4).

Do této skupiny strategie patří i poslední tah. Bílý hráč začíná a je sudý počet polí na desce, takže zakončovací tah, pokud nedojde k vynechání tahu, je na černém hráči. To je pro něj samozřejmě jistá výhoda, protože může převrátit velký počet kamenů na konci hry. Proto se bílý hráč musí snažit, aby počet vynechání tahů byl lichý.

4.3.5 Předvídavost

Toto není ani tak teoretická strategie, jako spíše schopnost daného hráče. Hráč si musí pokud možno představit o několik tahů dopředu, jakým směrem se daná hra bude odvíjet. Měl by tedy mít i malou dávku empatie, aby se vcítil do svého protivníka a věděl, zda bude v danou chvíli bránit, útočit nebo vyčkávat. Hráč s vyvinutou předvídavostí, má také usnadněny všechny ostatní taktické možnosti, které jsou zde uvedeny.

4.3.6 Zakončení

Existuje spousta způsobů, jak u dané hry postupovat ke konci partie. Na konci se totiž strategie různě mění. Některé kameny můžeme soupeři klidně přenechat, abychom jinde získali převahu. Nebo mu schválně nezpřístupníme určitá místa hrací desky. Toto uvádím jen jako několik málo příkladů z mnoha, aby si čtenář uvědomil, že na konci hry se těžiště strategie mírně mění.

5 Umělá inteligence a Reversi

5.1 Stavový prostor

Stavový prostor je v podstatě strom, jehož uzly jsou daným stavem modelu prostředí. Kořen tohoto stromu tvoří počáteční model. Pomocí přesně definovaných pravidel a akcí se snažíme z tohoto počátečního stavu dosáhnout stavu požadovaného, neboli cílového. Uzly jsou tedy stavy modelu a hrany jsou přechody (aplikované akce) mezi těmito stavy. Posloupnost použitých akcí, která zaručuje dosažení takového výsledku, se nazývá *plán*. *Plán* se tvoří pomocí metod k tomu určených a tyto metody jsou označovány jako *metody řešení úloh*.

Počáteční stav je popsán jednoznačně, ale cílový stav tuto vlastnost nemá. Někdy sice víme, jak má konec vypadat, ale někdy je určen pouze podmínkami, které musí splňovat, aby byl platným. Např. hra šachy. Nejčastěji uváděný příklad vztahující se k tomuto tématu. Podmínky pro cílový stav jsou 3. 1) Bílý král je v ohrožení a bílý hráč tomuto nemůže nikterak zabránit. 2) Černý král je v ohrožení a černý hráč tomuto nemůže nikterak zabránit. 3) Remíza. Nevíme, jak přesně budou figury na šachovnici rozestaveny. Víme pouze cílové podmínky. Kdybychom však věděli, jak přesně má šachovnice v cílovém stavu vypadat (nemusí to být konečný stav partie), tak by to bylo explicitní určení konečného modelu.

Metody řešení úloh, jak již bylo řečeno, slouží k vytváření a prohledávání stavového prostoru tak, abychom dosáhli cílového zadání. Všechny metody k tomu určené musí mít dvě vlastnosti, aby algoritmus mohl s jistotou oznámit, jestli řešení existuje a jaká cesta k němu vede. Tyto vlastnosti jsou: 1) musí docházet k pohybu ve stromu, ale tento pohyb nesmí být cyklický

2) k pohybu nesmí docházet náhodně – metoda musí postupovat podle jistých pravidel.

Jsou dva typy metod: informované a neinformované. Neinformované metody nemají žádné poznatky o daném problému. Takže na každý stav aplikují všechny pravidla, která mohou. A takto pokračují dále, dokud nezjistí dané řešení nebo že úloha řešení nemá. Strom těchto stavů však může být velmi rozsáhlý (např. počet možných prvních tahů jednoho hráče u šachu je 20, takže možností obou hráčů je 400 a to jen u prvních tahů) a v některých úlohách víme, že některé větve nemají šanci na úspěch. Pokud máme nějaké takové znalosti, můžeme aplikovat informované metody a tyto větve odříznout. Jestli je daná úloha konstruována tak, že můžeme vytvořit hodnotící funkci, můžeme pokračovat jen tou jednou cestou v grafu, která má největší šanci na úspěch. Tuto hodnotící funkci aplikujeme vždy na každý generovaný stav a vybereme nejpříznivější cestu. A jak vytvoříme takovou hodnotící funkci? Opět záleží na charakteru dané úlohy. Ale musí popisovat co nejvíce vlastností

dané úlohy. Např. pro šachy by mohla hodnotící funkce ohodnocovat figury (král má největší hodnotu, dáma menší, ... poslední je pěšec) a přítomnosti dané figury na šachovnici by se její hodnota sečetla s celkovou. Čím větší celkové ohodnocení pro nás a menší pro soupeře, tím lépe. Samozřejmě, že ohodnocení dané figury se může časem měnit. Pěšec, kterému chybí jen jedno pole, aby přešel celou hrací desku, bude mít zajisté větší ohodnocení, než jiný a „nepotřebný“. Detailní seznámení s aplikovanou hodnotící funkcí je v podkapitole 5.4.

5.2 Metoda minimax

Metoda minimax je asi nejčastěji používaná metoda u deskových her. V podstatě je to neinformovaná metoda prohledávání do hloubky, ale tato hloubka je omezena parametrem. V této hloubce se všechny stavy ohodnotí funkcí a poté se vybírá nejlepší cesta.

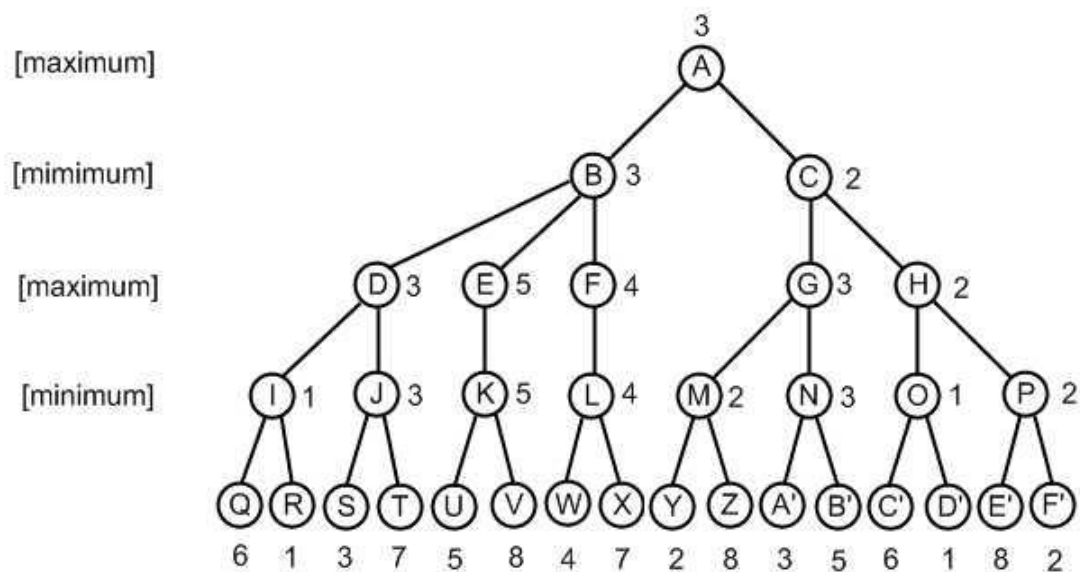
Procedura minimax funguje takto:

- 1) Ověříme, zda stav, kterým byla procedura minimax volána – dále jen STAV, je listem. Tedy jestli hra končí nebo jestli jsme již v potřebné hloubce zanoření. Pokud ano, ohodnotíme tento uzel a tuto hodnotu vrátíme. Pokud ne, pokračujeme dále.
- 2) Pokud je na tahu hráč, pro kterého je minimax volán, resp. pokud se nacházíme na liché úrovni stromu, generujeme všechny možné jeho tahy pro STAV a postupně pro ně voláme proceduru minimax. Z navracených hodnot vybereme maximum. Pokud je STAV stavem počátečním, tak si musíme i zapamatovat cestu k tomuto maximu.
- 3) Pokud se nacházíme na sudé úrovni stromu, tedy zda je na řadě protivník, tak také aplikujeme veškeré možné tahy pro STAV a postupně pro ně voláme proceduru minimax, ale z navracených hodnot vybereme hodnotu minimální.

Postupně tedy generujeme strom a až se nacházíme na úrovni listů, tak je ohodnotíme a rodič těchto listů poté dostane přidělení takové hodnoty, jaké odpovídá nejlepší ohodnocení (minimální na sudé úrovni nebo maximální na úrovni liché) jeho synů. Takto se postupuje stromem až ke kořenu.

Příklad 5.1

Předpokládejme, že se potřebujeme zanořovat do hloubky 4 a máme k dispozici hodnotící funkci, jejíž hodnoty se nachází v rozmezí od 1 do 9. Graf stavů je znázorněn na obrázku 5.1.



Obr. 5.1 Graf vzniklý pomocí metody minimax

Jak si může čtenář povšimnout, graf stavů není binární strom – tedy uzel nemusí mít jen dva synovské uzly. Získáme jej tak, že na daný stav (uzel) aplikujeme veškerá pravidla. To uděláme s každým uzlem, dokud se nenachází v požadované úrovni zanoření. Podle ohodnocení synů na páté úrovni (Q, R, S, T, U, V, W, X, Y, Z, A', B', C', D', E', F') se ohodnotí uzly na úrovni čtvrté – jejich otcové, (I, J, K, L, M, N, O, P) a to takto:

$$\begin{aligned}
 f(I) &= \min \{f(Q), f(R)\}, \\
 f(J) &= \min \{f(S), f(T)\}, \\
 f(K) &= \min \{f(U), f(V)\}, \\
 f(L) &= \min \{f(W), f(X)\}, \\
 f(M) &= \min \{f(Y), f(Z)\}, \\
 f(N) &= \min \{f(A'), f(B')\}, \\
 f(O) &= \min \{f(C'), f(D')\}, \\
 f(P) &= \min \{f(E'), f(F')\}.
 \end{aligned}$$

Pomocí takto ohodnocených uzlů můžeme ohodnotit jejich otce.

$$\begin{aligned}
 f(D) &= \max \{f(I), f(J)\}, \\
 f(E) &= \max \{f(K)\}, \\
 f(F) &= \max \{f(L)\}, \\
 f(G) &= \max \{f(M), f(N)\},
 \end{aligned}$$

$$f(H) = \max \{f(O), f(P)\}.$$

Systém ohodnocování se neustále opakuje (střídá se minimum a maximum).

Pokud se nachází uzel na liché úrovni, tak vybíráme maximum. Mluvíme tedy o maximalizující úrovni. Vybíráme maximum, protože je to nejlepší možný tah pro nás. Čím větší ohodnocení, tím lépe.

Pro sudé (minimalizující) úrovně platí přesný opak oproti lichým. Vybíráme tedy minimum a to proto, že se jedná o tah protihráče. Pokud tedy vybíráme minimum, domníváme se, že protihráč bude hrát pro něj nejlepší možnou cestou, tudíž pro nás nejhorší. Proto je ohodnocení nejnižší. V aplikaci je však zabudována i náhoda. Pokud najdeme lepší cestu, než jakou máme zatím uloženou (programově je minimax řešen rekurzí, takže si pamatujeme zatím nejlepší nalezenou cestu), tak máme šanci 70%, že půjdeme touto lepší cestou. Tato náhoda neplatí pro mód učitel a pro ostatní módy je přítomna z toho důvodu, aby počítač netáhl vždy stejně.

Pro úplnost uvedu i zbylé dvě úrovně k obrázku 5.1:

$$f(B) = \min \{f(D), f(E), f(F)\},$$

$$f(C) = \min \{f(G), f(H)\}.$$

Poslední ohodnocení se provádí u počátečního stavu. Takto bude počítač vědět, jaká větev celého stromu je pro něj nejvýhodnější.

$$f(A) = \max \{f(B), f(C)\}.$$

5.3 Prořezávání Alfa-beta

Generování všech možných stavů a jejich následné hodnocení může být někdy velmi náročné. Jak jsme viděli na obrázku 5.1, graf se velmi rychle rozrůstá a to většina uzlů má jen dva následovníky. Někdy může mít jeden uzel např. následovníků třeba 20. Pokud bychom generovali takovýto strom do hloubky 6, vygenerovali bychom 20^6 listů, což je 64 000 000. To už je velmi velký strom a procesoru to zabere to nemalý čas. Proto bylo nasnadě vymyslet nějakou jinou metodu nebo tuto metodu určitým způsobem urychlit. A tak vznikla metoda prořezávání alfa-beta. Tato metoda nám zaručuje, že se nebudeme zabývat generováním a vyšetřováním takových větví, které od stromu můžeme odříznout, protože určitě nemohou obsahovat lepší řešení, než které jsme už našli. Alfa řezy nám zaručí, že nedojde k zbytečnému vyšetřování hráče, pro kterého se minimax volal a beta řezy nám toto zabrání pro jeho protihráče. Pokud takto eliminujeme nějaký uzel např. v úrovni 2, tak celým jeho podstromem se už nebudeme zabývat. Což je velmi výhodné.

Účinnost prořezávání je však vždy jiná, podle toho, kde se nachází nejlepší řešení. Pokud je toto řešení, resp. cesta generována jako první, k dalšímu vyšetřování někdy ani nemusí dojít. Ale to je ideální příklad. Naopak, pokud je nejlepší cesta generována jako poslední, tak se muselo projít celým stromem, než jsme tento fakt zjistili. Většinou však k těmto extrémům nedochází. Takže často určitou část nemusíme prověřovat, ale tato část nemusí být příliš velká.

Procedura alfa-beta by se dala popsat takto:

Poprvé voláme proceduru s počátečním stavem, $\alpha = -\infty$ a $\beta = \infty$. Počítač nekonečno neumí a proto α musí být alespoň o 1 menší než je nejmenší možné ohodnocení hodnotící funkce a β musí být alespoň o 1 větší než je největší možné ohodnocení zmíněné funkce.

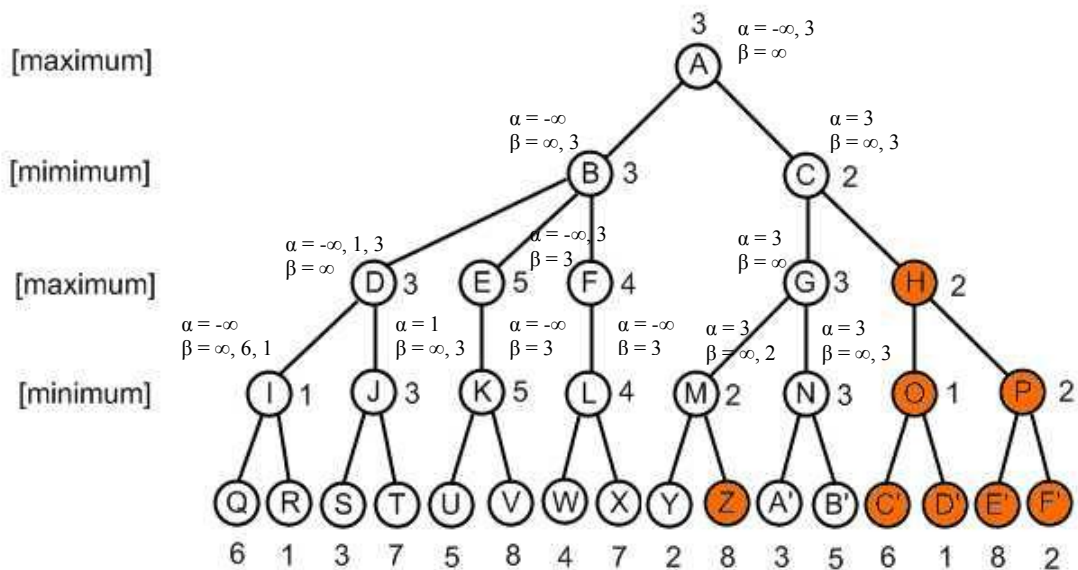
- 1) Pokud je stav, pro který byla momentálně procedura alfa-beta volána (dále jen STAV), stavem konečným – listem stromu (jestli hra končí nebo jestli jsme již v potřebné hloubce zanoření), tak vrať hodnotu STAVu a proceduru ukonči.
- 2) Pokud se STAV nachází na liché úrovni generovaného stromu, tak pokračuj. Jinak skoč na bod č. 5.

Hraje hráč, uzel je typu OR

- 3) Dokud je $\alpha < \beta$, tak volej proceduru alfa-beta pro další uzel, který jsme vygenerovali pomocí aplikovaného pravidla na STAV, a s momentálními hodnotami α a β . Vždy když se jeden uzel prošetří, tak porovnej navrácenou hodnotu s α a větší hodnotou tuto proměnnou přepiš.
- 4) Vrať hodnotu α a proceduru alfa-beta ukonči. Pokud se nacházíme na druhé úrovni stromu (hloubka zanoření = 1), tak vrať i tah, ať kořenový uzel ví, jaký tah má udělat, aby se dostal do této větve pokud bude vybrána jako nejlepší.

Hraje protihráč, uzel je typu AND

- 5) Dokud je $\alpha < \beta$, tak volej proceduru alfa-beta pro další uzel, který jsme vygenerovali pomocí aplikovaného pravidla na STAV, a s momentálními hodnotami α a β . Vždy když se jeden uzel prošetří, tak porovnej navrácenou hodnotu s β a menší hodnotou tuto proměnnou přepiš.
- 6) Vrať hodnotu β a proceduru alfa-beta ukonči.



Obr. 5.2 Prořezání alfa-beta

Navážeme na příklad 5.1 a na obrázek 5.1 a aplikujeme prořezávání alfa-beta. Graficky znázorněný výsledek je na Obr. 5.2.

Zavoláme proceduru alfa-beta pro počáteční stav A s hodnotami $\alpha = -\infty$ a $\beta = \infty$. Na tento stav aplikujeme určité pravidlo, tedy tah hráče, pro kterého je procedura volána. Tímto jsme se dostali do stavu B. Tento uzel také není uzlem listovým a proto zavoláme proceduru alfa-beta opět s danými parametry tak, jako by táhl protihráč. Takto se dostaneme přes uzel D (táhne hráč) až k uzlu I (táhne protihráč), kde opět zavoláme proceduru alfa-beta. Ta nám oznámí, že uzel Q je uzlem listovým a proto nám navrací jeho hodnotu, kterou určila hodnotící funkce, tedy hodnotu 6. Tato hodnota je menší než ∞ a proto se jí přepíše proměnná β a zavoláme proceduru alfa-beta z uzlu I pro další možný tah protihráče, ale nyní již s hodnotami $\alpha = -\infty$ a $\beta = 6$. Uzel R je opět listovým uzlem a jeho návratová hodnota je 1. Tato hodnota je menší než současný stav proměnné β , takže proměnnou β přepíšeme na hodnotu 1. Tímto už z uzlu I nemůžeme nikam jinam táhnout a proto vracíme hodnotu β , kterou porovnáme s otcovskou α . Ta je menší, takže ji přepíšeme. Uzel D má nyní již prošetřenu jednu větev a hodnota α se nyní rovná 1 a $\beta = \infty$. Aplikujeme další možný tah a tímto se dostaneme do uzlu J (s danou α, β). Zde po použití dalšího pravidla dostaneme z koncového uzlu S hodnotu 3. Tato hodnota je menší než současná β . Nastavíme tedy $\beta = 3$. Z další větve uzlu J získáme hodnotu 7, což je hodnota větší než současná β a proto je pro nás nezajímavá. Tímto způsobem bychom mohli pokračovat ještě přes uzly E, K, U, V, F, L, W, X. Ale žádných změn by se nedosáhlo, takže tyto uzly přeskočíme a budeme se věnovat pravé větvi počátečního uzlu A. Do této větve vstupujeme s hodnotami

$\alpha = 3$ a $\beta = \infty$. Zdůrazňuji, že k **vyšetřování dalších větví dojde jen pokud $\alpha < \beta$** . Po dalším hypotetickém tahu hráče se dostáváme do uzlu C. Po jednom z možných tahů protihráče do uzlu G. Opět použijeme jedno z možných pravidel a tímto se dostaneme do uzlu M. Po tahu protihráče (zavoláme proceduru alfa-beta pro M, $\alpha = 3$, $\beta = \infty$) získáme hodnotu 2. Touto hodnotou můžeme přepsat proměnnou β . Nyní je však $\alpha > \beta$ a proto nemusíme vyšetřovat ostatní možné varianty z uzlu M (v tomto případě se nebude prošetřovat jen uzel Z). Nyní bychom pokračovali další větví uzlu G. Ale v uzlech N, A', B' se žádné hodnoty nepřepíše a proto je můžeme přeskočit, když procedura by je prošetřila. Z uzlu G tedy vrátíme hodnotu α , která má v tuto chvíli hodnotu 3. Touto hodnotou se přepíše hodnota β v uzlu C, takže teď nám vznikla situace, kdy $\alpha = \beta$. Ale abychom mohli prošetřovat další větev, musí platit $\alpha < \beta$. To tedy neplatí a nemusíme procházet další podstrom od uzlu H včetně.

Jak vidíme z tohoto příkladu, prošli jsme 23 uzlů a 8 jsme vynechali. Samozřejmě se jedná jen o smyšlený příklad bez jakékoliv aplikace, ale demonstruje, že v tomto příkladu jsme mohli vynechat téměř 35% celkového grafu.

Hra Reversi má generované stromy mnohem větší, zvláště pak, když lze v aplikaci nastavit obtížnost od hloubky zanoření 1 až po 8. Průměrně je možné zahrát tah něco málo více než osmi způsoby. Takže u hloubky zanoření 8 a s 8 možnými různými tahy se strom rozroste na 16 777 216 listových uzlů. Pokud bychom tedy i tady nemuseli prošetřovat řekněme 35% (toto číslo se nedá vypočítat) nemuseli bychom se zabývat 5 872 026 uzly. A to už se na rychlosti tahů počítače opravdu projeví.

5.4 Hodnotící funkce

Hodnotící funkce, nazývaná také heuristická funkce nebo statická hodnotící funkce, určuje inteligentnost chování programu. Na co nám bude, když se budeme zanořovat například do hloubky 10, když nám potom hodnotící funkce vrátí zkreslené nebo špatné hodnoty? To by potom znamenalo, že veškeré průchody byly zbytečné, protože algoritmus (minimax, alfa-beta) bude vybírat jiné cesty než jaké jsou ideální.

Vytvořit hodnotící funkci však nemusí být vždy snadné. Pokud bychom chtěli vytvořit hodnotící funkci pro hru lišák (hrací pole 3x3, nachází se zde 8 kamenů a jedna mezera; kameny musí být poskládány v určitém sledu), tak by hodnotící funkce pravděpodobně vracela počet kamenů, které jsou na svých pozicích. U této hry je však známo, jak vypadá cílový stav. Zatímco cílový stav u hry Reversi je určen pouze podmínkou, že ani jeden z hráčů nemůže udělat svůj tah. Hrací deska (8x8) v tomto okamžiku může vypadat přesně 717 897 987 691 852 588 786 633 (tedy $718 \cdot 10^{21}$) způsoby, pokud počítám, že nejmenší možná kombinace tahů, která hru ukončí, je 10. Triviální hodnotící

funkce pro hru Reversi by mohla vracet takovou hodnotu, jaké odpovídá poměr počtu přítomných kamenů od hráče s bílými kameny a hráče s kameny černými. Pokud bychom však využívali takovou funkci, zjistili bychom, že není příliš dobrá, protože každý jeden kámen by brala jako hodnotu „1“. Takže někdy by si program vybral cestu grafem takovou, která mu sice momentálně obsadí o pár kamenů více, ale v dalších tazích o ně může velmi rychle přijít, protože je nemá na strategických pozicích. Hodnotící funkce v Reversích musí tedy zohledňovat jak počet kamenů daných barev, tak i pokud možno co nejvíce strategických metod, které jsou popsány v kapitole 4.3.

	A	B	C	D	E	F	G	H
1	30	-2	12	12	12	12	-2	30
2	-2	-3	0	1	1	0	-3	-2
3	12	0	5	5	5	5	0	12
4	12	1	5	5	5	5	1	12
5	12	1	5	5	5	5	1	12
6	12	0	5	5	5	5	0	12
7	-2	-3	0	1	1	0	-3	-2
8	30	-2	12	12	12	12	-2	30

Tab. 5.1 Hodnotící matice na začátku hry

V Tab. 5.1 lze vidět hodnotící matici tak, jak vypadá na začátku hry. Tabulka má barevně rozlišené pole, kde světlé barvy znázorňují malé hodnoty a tmavší barvy hodnoty nejvyšší.

Nyní vysvětlím, proč jsou hodnoty pole různé. Je to právě dané zohlednění strategie. Pokud hráč má umístěn svůj kámen např. na poli D3, tak se mu tento kámen počítá jako kamenů 5. Toto počítání však na celkový výsledek nemá žádný vliv. Pokud hra skončí, tak má každý bodů jen tolik, kolik má na desce svých kamenů. Tato tabulka nám jen znázorňuje, hodnotící funkci a nikterak nezasahuje do pravidel hry. Hráči se tedy

snaží být nejdéle ve středu hřiště a nevkročit do kritické oblasti (Obr. 4.4 **Kritická oblast**), protože by svému protihráči nabídli okraj. Je to zajištěno většími hodnotami polí ve středu hřiště a o poznání menšími ve sloupcích B a G a řádcích 2 a 7 (pokud uvažujeme herní desku 8x8). Ale pokud je k tomu některý z hráčů donucen, tak raději by měl svůj kámen umístit od rohu hrací desky co nejdále. Tak bude mít protihráč z jeho tahu nejmenší profit. Proto je u předrohového pole hodnota dokonce záporná, zatímco u polí, které jsou před okrajem, ale u středu hřiště je tato hodnota rovná jedné.




Hodnotící funkce se však skládá z mnoha podmínek a Tab. 5.1 **Hodnotící matice na začátku hry** tabulka nám jen znázorňuje, hodnotící funkci a je pouze jejím výřezem - tedy dosažení do podmínek hodnotící funkce takových hodnot, že dostaneme matici na počátku hry. Celková hodnotící funkce zapsaná v tabulce se nachází v Tab. 5.2 **Hodnotící funkce**.

	A	B	C	D	E	F	G	H
1	30	$\frac{-7}{3/1}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-7}{3/1}$	30
2	$\frac{-7}{3/1}$	$\frac{-7}{3}$	$0/3$	$1/3$	$1/3$	$0/3$	$\frac{-7}{3}$	$\frac{-7}{3/1}$
3	$\frac{-3}{12}$	$0/3$	5	5	5	5	$0/3$	$\frac{-3}{12}$
4	$\frac{-3}{12}$	$1/3$	5	5	5	5	$1/3$	$\frac{-3}{12}$
5	$\frac{-3}{12}$	$1/3$	5	5	5	5	$1/3$	$\frac{-3}{12}$
6	$\frac{-3}{12}$	$0/3$	5	5	5	5	$0/3$	$\frac{-3}{12}$
7	$\frac{-7}{3/1}$	$\frac{-7}{3}$	$0/3$	$1/3$	$1/3$	$0/3$	$\frac{-7}{3}$	$\frac{-7}{3/1}$
8	30	$\frac{-7}{3/1}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-3}{12}$	$\frac{-7}{3/1}$	30

Tab. 5.2 Hodnotící funkce

Na této tabulce lze vidět, jakých veškerých hodnot mohou daná pole dosahovat. Ale jelikož by bylo poměrně složité vysvětlovat na celé tabulce, jaká pole jsou závislá na kterých polích, tak v dalších tabulkách jsou udělány přehlednější výřezy a podrobněji jsou vysvětleny jejich vazby mezi sebou.

V následujících tabulkách (5.3 – 5.7) se bude uplatňovat tato legenda:

	Pole je vysvětlováno. Je závislé na jiném poli.
	Toto pole ovlivňuje pole jiné.
	Kombinace obou předchozích variant.

V podkapitole 4.3.1 jsme se dočetli o jedné z podstrategií ohledně hry Reversi, která se zabývá nezpřístupněním rohů. Tato strategie je implementována v hodnotící funkci a to tak, že pole, která jsou před rohovým herním políčkem, jsou hodnotově znevýhodněná.

Tabulka 5.3 se zabývá poli, která se nacházejí od rohového pole směrem do středu hrací desky. Tato pole mohou nabývat hodnot -7 nebo 3 , podle toho, zda je obsazen roh, u kterého se toto pole nachází. Jestli je roh prázdný, tak hodnota je rovna -7 . Je to velký postih, protože je potom pravděpodobné, že soupeř pomocí tohoto pole obsadí roh a s tímto tahem i toto předlohové pole. Jestli však už

je roh obsazen, tak k žádnému postihu nedochází. Násobný koeficient tohoto pole bude dosahovat čísla 3.

Hodnocení polí nacházející se na okraji před rohem je znázorněno v tabulce 5.4. Pokud je roh prázdný, je toto pole ohodnoceno -7. Pokud je obsazen, jsou dvě možnosti. Roh máme v moci my. Hodnota pole je tedy 3. Druhá možnost je, že roh je obsazen soupeřovým kamenem a to by poté pole nabylo hodnoty 1.

	A	B	C	D	E	F	G	H
1	30							30
2		-7 / 3					-7 / 3	
3								
4								
5								
6								
7		-7 / 3					-7 / 3	
8	30							30

Tab. 5.3 Hodnocení předrohového pole

	A	B	C	D	E	F	G	H
1	30	-7 / 3 / 1					-7 / 3 / 1	30
2	-7 / 3 / 1							-7 / 3 / 1
3								
4								
5								
6								
7	-7 / 3 / 1							-7 / 3 / 1
8	30	-7 / 3 / 1					-7 / 3 / 1	30

Tab. 5.4 Hodnocení okraje před rohem

V následující podkapitole týkající se strategií (4.3.2) jsme se zabývali okrajem. Okraj je totiž poměrně důležité taktické místo, ze kterého se dá dobře útočit a tím reversovat soupeřovy kameny. Proto soupeři nesmíme (pokud to jde) okraj zpřístupnit. Pokud to ovšem již není možné a musíme mu otevřít cestu k okraji, tak je pro nás nejvýhodnější to udělat přes střed hrací desky. Tímto středem se zabývá tabulka 5.5. Tato pole se nachází pouze na deskách o rozměrech 8x8 a 10x10. Nedochozí u nich k žádnému postihu, ale zvýhodnění je pouze u jednoho ze dvou možných stavů. A to, pokud je již roh obsazen. Poté víme, že tímto tahem soupeři roh nezpřístupníme, protože by svůj kámen neměl kam umístit. Tedy může mít jednu volnou pozici na umístění svého hracího kamene, protože stačí, aby okraj byl zaplněn pouze ze dvou třetin přilehlého pole. Jedna třetina je zanedbatelná, protože pokud má soupeř obě zbývající pole v jeho dispozici, tak mu tuto řadu pouze prodloužíme, ale stále tato řada může být reversována z rohu. V opačném případě, že obě pole budou obsazena našimi kameny, případný soupeřův kámen bychom ihned zajali. Samozřejmě se může stát, že každé z obsazených polí bude od jiného hráče. To bychom se však nemuseli rozmyšlet nad touto variantou tahu.

Např. pole B1 je závislé na polích A3, A4 a A5. Pro zvýraznění závislostí není vykresleno pole B5 a „jeho“ pole A6 a analogicky E2 s F1. Další pole jsou již všechna vykreslena a zvýrazněna.

	A	B	C	D	E	F	G	H
1			-3 / 12	-3 / 12	-3 / 12	-3 / 12		
2				1/3	1/3			
3	-3 / 12							-3 / 12
4	-3 / 12	1/3					1/3	-3 / 12
5	-3 / 12	1/3					1/3	-3 / 12
6	-3 / 12							-3 / 12
7				1/3	1/3			
8			-3 / 12	-3 / 12	-3 / 12	-3 / 12		

Tab. 5.5

Pokud nemáme jinou možnost a musíme okraj soupeři zpřístupnit a nemůžeme to udělat přes střed hřiště, tak jediné, co nám zbývá je udělat to blíže k rohu (snažíme se nedat protihráči možnost získat roh). O tomto příkladu vypovídá tabulka 5.6, která znázorňuje, jakých hodnot budou nabývat pole, kterých se to týká. Tato pole mohou mít index, který se bude počítat se sumou ostatních hodnot na polích jenž obsahují stejnou barvu, buď 0 nebo 3. Hodnotu 3 bude dané hrací políčko obsahovat, pokud pole jemu přilehlé na okraji bude obsazené. Jestli toto pole (např. pro C2 je toto pole C1) bude zatím prázdné a my budeme chtít přeci jen položit svou hrací figuru na zmiňované pole, jenž leží v kritické oblasti, tak musíme počítat s tím, že se nám žádná hodnota nezapočítá, protože ač budeme mít kámen v tomto místě umístěn, tak jeho index se bude rovnat nule. Je to z toho důvodu, že se již velmi přibližujeme k rohu, který má velkou strategickou sílu a nemůžeme si dovolit, aby jej vlastnil protivník.

	A	B	C	D	E	F	G	H
1			-3 / 12			-3 / 12		
2			0/3			0/3		
3	-3 / 12	0/3					0/3	-3 / 12
4								
5								
6	-3 / 12	0/3					0/3	-3 / 12
7			0/3			0/3		
8			-3 / 12			-3 / 12		

Tab. 5.6

Předchozí tabulky se zabývaly zpracováním strategií rohů a okrajů pomocí hodnotící funkce. Avšak ještě jsem uvedl, že taktických manévrů je více. Ještě zbývá pohyblivost, rovnováha, předvídavost a zakončení.

Pohyblivost

Ta je také řešena pomocí heuristické funkce a to tak, že pokud ve stromu narazíme na větev, která je ukončena tím, že soupeř nemůže táhnout, tak tuto větev ohodnotíme číslem 697. Maximální možná hodnota součtu veškerých polí hrací desky dá totiž součet 696. Téměř vždy je pro nás výhodné, když táhneme vícekrát za sebou, aniž bychom pustili k tahu protihráče. Ovšem jestli se v generovaném stromu stavů nachází větev, jejíž list určuje stav takový, že nemůžeme táhnout my, tak tento list ohodnotíme indexem -697, protože to je pro nás nejméně výhodná pozice do které bychom se neměli pokud možno dostat. Aplikace tedy nebere jako rozdíl to, zda protihráč má možnost táhnout jen na dvě pole nebo na 10. To je málo závažné a téměř zbytečné. O část pohyblivosti se už postará generovaný graf pomocí minimaxu a nemusíme jí řešit v sekci hodnotící funkce.

Rovnováha, předvídavost, zakončení

Hodnotící funkce rovnováhu neřeší. Rovnováha je řešena jen pomocí minimaxu, tedy vybrání vhodného tahu v dané situaci. Vždy se snažíme zamezit soupeři v jeho tahu a táhnout tak sami několikrát po sobě.

Předvídavost se týká spíše člověka. Počítač má přesně definovanou hloubku zanoření pro algoritmus minimax, resp. pro prořezání alfa-beta a přesně tuto hloubku dodržuje. Nevybere si jednu „příznivější“ cestu a potom se na ni nezaměří a neprojde ji do větší hloubky. Jeho předvídavost je tedy staticky stanovena. Pokud má člověk vědomosti o tom, jak je naprogramována hodnotící funkce a zvládne se pokusit pomocí této vědomosti odhadnout, kam bude táhnout počítač, tak je na jeho tahy už připraven a může je potom s větší lehkostí zvládnout.

Poslední taktická sekce se nazývá zakončení. Jak jsem již poznamenal výše, je mnoho typů zakončení. Avšak jejich realizace již potřebuje dobré znalosti hry a jejich správná volba v dané situaci je nezbytná. Je to spíše pro profesionální hráče a já takový hráč rozhodně nejsem a proto ani nevím, jak přesně se tyto taktiky propočítávají. Jejich implementace proto v aplikaci chybí. Snad tímto nedostatkem hráč nebude ošizen o kvalitu hry, ale to ať si každý zodpoví sám. Avšak i na konci hry jsou stále propočítávány veškeré možné tahy, takže počítač i ve finální části hry by měl táhnout tak, jak je to pro něj nejvýhodnější.

Je však ještě jedna situace hry, která je zohledněna v hodnotící funkci a to je táhnutí na kraj. Člověku je totiž jasné, že pokud táhne vedle protivníkova kamene a má za sebou volné pole, tak že ho protihráč přeskochí. Nejzávažnější by tato chyba byla na okraji. Musíme tedy měnit i hodnoty okraje v závislosti na tom, zda je vedle potenciálního tahu pole s protivníkovým kamenem. Pokud bychom tedy umísťovali náš kámen vedle protivníkova kamene a tato akce by se odehrávala na okraji, musíme počítat s postihem -3. Pokud je vedle našeho nově umístěného kamene další náš kámen nebo prázdné pole, tak potom nás čeká bonus s indexem 12. Tabulka 5.7 nám detailně znázorňuje, pro jaká pole toto platí. V prvním řádku a sloupci je zakreslen vždy jeden příklad toho, jaká pole jsou kterými ovlivněny. V posledním řádku a sloupci jsou pak ukázána veškerá pole, pro které toto platí. Jestli tedy chceme umístit svůj hrací kámen na pole E1, tak index tohoto pole bude 12, ale jen v tom případě, pokud ani na jednom ze sousedních polí, tj. D1 a F1, nebude soupeřův kámen.

	A	B	C	D	E	F	G	H
1		-7 / 3 /	-3 / 12	-3 / 12	-3 / 12	-3 / 12	-7 / 3 /	
2	-7 / 3 /							-7 / 3 /
3	-3 / 12							-3 / 12
4	-3 / 12							-3 / 12
5	-3 / 12							-3 / 12
6	-3 / 12							-3 / 12
7	-7 / 3 /							-7 / 3 /
8		-7 / 3 /	-3 / 12	-3 / 12	-3 / 12	-3 / 12	-7 / 3 /	

Tab. 5.7 Hodnocení okraje

5.5 Strojové učení

Umělá inteligence je schopna táhnout na jiné pole, když bude vědět, že v některé z minulých her se vyskytla stejná sekvence tahů a počítač prohrál. Má tedy větší šanci, že vyhraje, protože se této posloupnosti tahů už vyvaruje. Hra počítače je tedy lepší než v předchozích hrách. A toto chování můžeme nazvat strojovým učení.

Implementace tohoto strojového učení ve hře Reversi je poměrně jednoduchá. Spustí se jen když máme více než jednu možnost táhnout na stejně dobře ohodnocené pole. Takže vždy potáhneme

na pole s nejlepší hodnotou a pokud těchto polí bude více, zeptáme se, jaké je výhodnější podle předešlých her.

Základem strojového učení je mít s čím srovnávat tahy. Musíme si tedy ukládat veškeré partie. Toto ukládání se v aplikaci děje jen v obtížnosti nazvané *Učitel*, protože u tohoto módu si uživatel nemůže nastavit hloubku zanoření pro minimax. Kdyby toto mohl udělat, tak by učení bylo velmi zkreslené, protože bychom mohli učit počítač s hloubkou zanoření 2 a poté nastavit tuto hodnotu na 7, kde by se umělá inteligence řídila i podle těchto méně promyšlených partií. Proto musí být hloubka nastavena staticky, aby docházelo k pozitivnímu efektu učení. Hloubka je volena na hodnotu 6.

Takže pokud máme uložené některé partie, musíme k nim ještě přiřadit určité informace. Potřebujeme vědět, za jakou barvu hrál člověk v té dané partii a kdo vyhrál. A nyní už máme veškeré potřebné informace a můžeme porovnávat s předchozími partiemi. Tahy porovnáваме jen u těch, ve kterých se shoduje barva kamenů, s kterými hraje člověk nyní, a barva kamenů, které ovládal člověk v uložených hrách. Jestli se sekvence tahů (v momentální a uložené partii) shodují, tak o jednu zvýšíme počet v proměnné *proher* nebo *výher*, podle toho, zda v dané uložené hře vyhrál počítač nebo člověk. Takto projdeme veškeré uložené hry a na konci budeme mít počet proher a počet výher (počítače). Pokud je výher více než proher, můžeme pokračovat v myšleném tahu. Jestli ne, tak daný tah zavrhneme.

Nejlepší na pochopení bude asi názorný příklad. Začneme novou hru na desce o velikosti 6x6 polí a budeme chtít táhnout jako druzí, tedy jako černý hráč. Počítač má 4 možné stejně kvalitní první tahy. Pokud budeme mít záznamy o těchto hrách:

- 1) černá prohra D5 E3 D2 C1 C2 ...
- 2) černá prohra E4 E3 D2 C1 C2 ...
- 3) černá výhra D5 E3 D2 C1 C2 ...
- 4) bílá prohra D5 E3 D2 C1 C2 ...
- 5) černá výhra D5 E3 D2 C1 C2 ...

, kde černá/bílá značí barvu kamenů s kterými hraje člověk a výhra/prohra se vztahuje k počítači, a počítač by zvažoval, zda umístit kámen na souřadnici D5, tak by se podíval na tyto záznamy. U prvního vše souhlasí, takže prohry = 1. U druhé nesouhlasí tah, takže přejde ke třetí. Zde se potřebné informace také shodují a proto výhry = 1. U čtvrté hry nesouhlasí barva – přeskočíme tuto hru. A pátá vyhovuje, takže výhry = 2. Výher je více než proher a proto můžeme prohlásit, že tah na souřadnici D5 může být úspěšný. Jestli by však pátý záznam minulých her nebyl nebo by vypadal takto:

- 6) černá prohra D5 E3 D2 C1 C2

, museli bychom tah na D5 zahrnout, protože by proher bylo více než výher. Takže bychom táhli na další pole, které nám algoritmus nabídne. V tomto případě by algoritmus narazil na pole B3, které nemá žádné informace o minulých hrách, takže bychom táhli na toto pole.

Popsaný příklad nemá žádné opodstatnění. Samozřejmě, že je jedno, kam bude umístěn první tah, protože hra bude pouze symetricky otočena. Má pouze znázorňovat, jak je dané strojové učení implementováno.

5.6 Ověření inteligence

Program je také vybaven síťovou vrstvou, přes kterou může komunikovat s managerem deskových her, který udělal Jiří Kusák jako svou bakalářskou práci také v tomto školním roce. Pomocí tohoto managera může umělá inteligence implementovaná v mém programu soupeřit s umělou inteligencí úplně jinou. Bohužel manager má nastavitelnou volbu toho, kdo bude začínat, což je nerespektování oficiálních pravidel, podle kterých je tato hra implementována. Takže pokud bude tento test probíhat, musí se uskutečnit tak, že bude v managerovi začínat hráč s černými kameny.

6 Závěr

Bakalářská práce se věnovala umělé inteligenci, která je aplikovaná na deskové hře Reversi. Vysvětlili jsme si, co je a čím se zabývá umělá inteligence, a jak je aplikovatelná na počítačové hry. Do větších detailů jsme se potom seznámili, jak lze umělou inteligenci použít pro tahy počítače u deskové hry Reversi.

Umělá inteligence naprogramovaná pomocí metody minimax s prořezáváním alfa-beta podle mého názoru hraje docela dobře. Rozhodně není neporazitelná, ale průměrný hráč musí taktizovat a přemýšlet, aby program porazil. Pokud k tomuto ještě přidáme počítačové učení, které je obsaženo v obtížnosti *Učitel*, tak se hra pro člověka opět stíží. Zvláště pak, pokud tuto aplikaci bude využívat jeden uživatel, protože každý má určitý typ strategie (ofenzíva, vyčkávání, defenzíva...) a tu se počítač naučí a bude pro něj lehčí tohoto hráče porazit.

Zlepšení tohoto programu je bezpochyby možné. A to minimálně čtyřmi způsoby:

- 1) **Hardwarovými možnostmi.** Je to cesta hrubé síly, což znamená, že bychom mohli prohledávat strom do větší hloubky za stejný čas, pokud bychom měli k dispozici lepší hardwarové prostředky.
- 2) **Paralelní prohledávání stromu.** Pokud bychom byli schopni metodu minimax paralelizovat, výsledek této metody by byl mnohem dříve, takže bychom mohli prohledávat opět do větší hloubky. Bohužel nevím, jak by se tato možnost dala realizovat a proto jsem minimax naprogramoval rekurzivně.
- 3) **Dynamizované hodnocení polí.** Hodnotící funkce nám sice zajišťuje, že každé pole na hrací desce je hodnoceno podle jeho důležitosti, ale hodnoty mohou být velmi zkreslené. Jsou stále stejné – statické. Pokud bychom zajistili určité propočítání, jaké indexy by měly být na jakém poli, tak bychom dostali ideální hodnotící funkci a počítač by tak hrál nejlépe, jak by bylo možné.
- 4) **Jiné metody implementace.** Toto platí pro mód *Učitel*. Ten by mohl být naprogramován pomocí jiných metod, např. neuronových sítí.

Literatura

- [1] A.L.I.C.E. *A.L.I.C.E. Artificial Intelligence Foudation*. [Online] [Citace: 7. Květen 2007.]
<http://www.alicebot.org/mailling-lists.html>.
- [2] Argument čínského pokoje. *Wikipedia*. [Online] Květen 2006. [Citace: 7. Květen 2007.]
http://cs.wikipedia.org/wiki/Argument_čínského_pokoje.
- [3] Artificial intelligence. *Wikipedia*. [Online] Březen 2007.
http://en.wikipedia.org/wiki/Artificial_intelligence.
- [4] **Mařík, V., Šepáková, O. a Lažanský, J.** *Umělá inteligence (1)*. Praha : Academia, 1993.
- [5] **Svršek, J.** Umělá inteligence. *Natura*. [Online] Březen 2004. [Citace: 7. Květen 2007.]
<http://natura.eri.cz/natura/2004/3/20040301.html>.
- [6] Turingův stroj. *Wikipedia*. [Online] Duben 2007. [Citace: 7. Květen 2007.]
http://cs.wikipedia.org/wiki/Turingův_test.
- [7] **Zbořil F., Zbořil F.** *Základy umělé inteligence*. Brno : VUT FIT, 2006.

Seznam příloh

Příloha 1. CD