

Knihovna logických her v prostředí Mathematica

Tomáš Trčka

Bakalářská práce
2014



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Trčka**
Osobní číslo: **A11272**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Knihovna logických her v prostředí Mathematica**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Seznamte se s různými logickými úlohami a problematikou jejich řešení.
3. Vyberte vhodné řešitelné úlohy.
4. Implementujte tyto algoritmy v prostředí Mathematica.
5. Vytvořte ucelenou knihovnu v prostředí Mathematica.
6. Prostudujte možnosti webové implementace pomocí SW webMathematica.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KOLEKTIV autorů. McGraw-Hills Conquering LSAT Logic Games, Third Edition.: McGraw-hill, 2010. ISBN 9780071717878.
2. ZEGARELLI, M. LSAT Logic Games For Dummies. 1st ed.: Wiley, 2010. ISBN 9780470590140.
3. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kolektiv Umělá inteligence (3), Nakladatelství Academia, 2001. ISBN 978-80-200-0472-6.
4. MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kolektiv Umělá inteligence (5), Nakladatelství Academia, 2007. ISBN 978-80-200-1470-2.
5. MAJER, O., A. V. PIETARINEN AND T. TULENHEIMO Games: Unifying Logic, Language, and Philosophy: Unifying Logic, Language, and Philosophy. 1st Edition: Springer, 2009. ISBN 9781402093746.
6. ZELINKA, I. Umělá. inteligence - hrozba nebo naděje. BEN - technická literatura, 2003. ISBN 80-7300-068-7.
7. GARDNER, M. The Colossal Book of Mathematics. Norton, 2001. ISBN 978-0-393-02023-6.

Vedoucí bakalářské práce:

doc. Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

28. února 2014


Termín odevzdání bakalářské práce:

13. června 2014

Ve Zlíně dne 28. února 2014


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- Že odevzdaná verze diplomové/bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Bakalářská práce si klade za cíl seznámit čtenáře s pravidly, historií, variantami a různými metodami řešení vybraných logických úloh. Úvod teoretické části je věnován problematice logiky jako takové, zejména jejímu praktickému (ne)využívání v běžné komunikaci moderní lidské společnosti. Dále jsou charakterizovány hlavolamy sudoku, zebra puzzle (známy též jako Einsteinovy hádanky) a obrázkové puzzle. Následně jsou zváženy možnosti webové implementace těchto her naprogramovaných v prostředí Mathematica. Praktická část jednak podrobně popisuje vytvořené aplikace z pohledu uživatele, a jednak nastiňuje hlavní myšlenky klíčových úseků programového kódu.

Klíčová slova: Mathematica, logické úlohy, sudoku, zebra puzzle, puzzle

ABSTRACT

The bachelor's thesis aims to acquaint the reader with the rules, history, variants and different methods of solving selected brain teasers. The beginning of the theoretical part is devoted to the logic itself, especially its practical (not) use within the normal communication of modern human society. Furthermore, puzzles such as sudoku, zebra puzzle (also known as Einstein's riddle) and jigsaw puzzle are characterized. Then, the possibility of implementing these games programmed in Mathematica into the web environment is considered. The practical part describes created applications from a user's perspective in detail, and also outlines the main ideas of key sections of code.

Keywords: Mathematica, brain teaser, sudoku, zebra puzzle, jigsaw puzzle

Rád bych poděkoval své nejbližší rodině za psychickou i materiální podporu během studia a vedoucímu doc. Ing. Romanu Šenkeříkovi, Ph.D. za cenné podněty, rady a připomínky při zpracovávání bakalářské práce.

„Hra je jeden z nejefektivnějších způsobů, jak zjednodušit život. Přesně to jsme dělali jako děti, ale v dospělosti jsme si hrát zapomněli.“

Albert Einstein

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 LOGIKA	11
1.1 FORMÁLNÍ LOGIKA	11
1.2 NEFORMÁLNÍ LOGIKA.....	11
1.2.1 Aristotelova logika	11
1.2.2 Omyly argumentace v lidské řeči.....	12
2 LOGICKÉ HRY	16
2.1 SUDOKU	16
2.1.1 Historie	16
2.1.2 Matematické pozadí	17
2.1.3 Metody řešení.....	17
2.1.3.1 Naked Singles (NS, zkráceně také Singles, S).....	19
2.1.3.2 Hidden Singles (HS)	20
2.1.3.3 Backtracking (BT, česky nejčastěji metoda zpětného vyhledávání) ...	20
2.1.4 Varianty	21
2.2 ZEBRA PUZZLE	21
2.2.1 Originální text hlavolamu	21
2.2.2 Varianty.....	22
2.2.3 Metody řešení.....	23
2.2.3.1 Metoda hypotéz.....	23
2.2.3.2 Metoda párování	24
2.2.3.3 Vylučovací metoda	24
2.2.3.4 Srovnání metod	25
2.3 PUZZLE.....	25
2.3.1 Historie	26
2.3.2 Varianty.....	26
2.3.3 Světové rekordy	27
3 MOŽNOSTI WEBOVÉ IMPLEMENTACE	29
3.1 WEBMATHEMATICA	29
3.2 CDF	29
3.3 SROVNÁNÍ	29
II PRAKTICKÁ ČÁST	31
4 SUDOKU	32
4.1 UŽIVATELSKÝ MANUÁL.....	32
4.1.1 Mřížka	32
4.1.2 Tlačítka.....	33
4.1.3 Zaškrtávací políčka	34
4.1.4 Rozevírací seznam	35
4.2 PROGRAMÁTORSKÝ MANUÁL	36
4.2.1 Deterministické metody řešení.....	36
4.2.2 Stochastické metody řešení	36
4.2.3 Metoda pro vytváření zadání.....	37

5	ZEBRA PUZZLE	38
5.1	UŽIVATELSKÝ MANUÁL.....	38
5.1.1	Vytváření zadání pro uživatele.....	38
5.1.2	Řešení vytvořených zadání uživatelem.....	39
5.1.3	Řešení zadání od uživatele	41
5.2	PROGRAMÁTORSKÝ MANUÁL	43
5.2.1	Metoda řešení	43
5.2.2	Metoda vytváření zadání	44
6	PUZZLE	45
6.1	UŽIVATELSKÝ MANUÁL.....	45
6.2	PROGRAMÁTORSKÝ MANUÁL	48
	ZÁVĚR	49
	SEZNAM POUŽITÉ LITERATURY.....	51
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	54
	SEZNAM OBRÁZKŮ	55
	SEZNAM TABULEK.....	56
	SEZNAM PŘÍLOH.....	57

ÚVOD

Již od základní školy jsem imponován všemožnými matematickými, logickými a programátorskými hlavolamy, což vyústilo v samostatnou nebo kolektivní účast na matematických olympiádách pro základní i střední školu, Internetové matematické olympiádě, matematické soutěži Náboj, soutěži pro „interdisciplinárně soutěživé bytosti“ Intersob, „internetové logické soutěži“ Interlos, internetové soutěži MathRace nebo informaticko-ekologickém soustředění K-SCUK, z nichž některých se stále pravidelně účastním.

Hlavní motivací pro volbu tohoto tématu pro mě byla touha vyzkoušet si logické úlohy z druhé strany – nevymýšlet řešení, ale vytvářet zadání. Před generováním úloh je samozřejmě nutné zrealizovat efektivního řešitele pro kontrolu existence řešení, případně ověření jeho unikátnosti, který musí pracovat velmi rychle, protože jej během vytváření zadání budeme volat velmi často.

Vývojovým prostředím byl zvolen SW Wolfram Mathematica, se kterým jsem byl seznámen na vysoké škole, a ze všech zde probíraných programovacích jazyků na něj byl kladen největší důraz. Mezi jeho výhody patří obrovská knihovna integrovaných funkcí, které pokryjí potřeby začínajícího i profesionálního programátora, výborně zpracovaná dokumentace v rámci „offline“ nápovědy samotného programu i „živé“ komunity na portále <http://mathematica.stackexchange.com/>, dále velká spolehlivost, vysoký výkon a značné množství vestavěných funkcí pro práci s grafickým uživatelským rozhraním. Největším nedostatkem je poměrně obtížná přenositelnost vytvořené aplikace oproti např. jazyku C vytvářejícím spustitelné „exe“ soubory, protože programy SW Mathematica lze spustit buď v jeho plné verzi, anebo v samostatném CDF přehrávači, jehož základní verze je sice zdarma, ale nefunguje na všech platformách.

Logické hry byly vybrány tak, aby obsáhly úlohu s čísly, slovy i obrázky. Luštění sudoku a skládání puzzle jsou mi blízké, protože se jim věnuji už od dětství. Uživateli poslouží spíše jako prostředek pro odreagování se či relaxaci, zatímco aplikace pro řešení zebra puzzle může posloužit lidem připravujícím se na české zkoušky SCIO či zahraniční LSAT.

I. TEORETICKÁ ČÁST

1 LOGIKA

Slovo „logika“ vychází z řeckého „logos“, které se nejvíce blíží českému výrazu „řeč“. Ve starověku byla logika chápána jako věda o řeči, která spolu s etikou a fyzikou tvořila celou tehdejší vědu zvanou filozofie.[1]

Dnes není možné takto obecný a složitý pojem definovat jednoznačně, proto jej různé zdroje vysvětlují různými způsoby. Je třeba také rozlišovat mezi vědním oborem a logikou jako takovou. Mně osobně nejbližší formulací je „metoda správného myšlení“. Vědní oblast se pak těmito metodami zabývá.[1]

Logika od svého vzniku souvisí s řečí, s vyjadřováním myšlenek, s myšlením samotným. Tento způsob uvažování je člověku vrozeným charakteristickým znakem, není potřeba se mu učit. Lidé dokáží uvažovat logicky, přestože nikdy žádným speciálním kurzem logiky neprošli.[1]

1.1 Formální logika

Jedná se o specifickou matematickou disciplínu, která toho s řečí nemá mnoho společného. Představuje jakousi umělou gramatiku, která používá přísně matematické vyjadřování a v jistém smyslu byla vytvořena zejména pro účely matematiky.[2]

Formální logika představuje nesmírně vyspělou vědu, která dokázala odhalit některé hluboké matematické zákonitosti (jako např. že matematiku nelze vystavět čistě axiomaticky, tj. že existují problémy, které není možné z axiomů odvodit, a je dokonce dokázáno, že takovéto axiomy nelze vůbec nalézt).[2]

1.2 Neformální logika

Toto odvětví logiky lze dále rozdělit na dvě podoblasti. První je blízká Aristotelově pojetí logiky, ta druhá se více zabývá argumentací v lidské řeči, respektive logickým zdůvodňováním častých chyb v lidské argumentaci.[1]

1.2.1 Aristotelova logika

Aristoteles se v podstatě taky k logice stavil formálně, nicméně jeho výroková a predikátová logika zůstává přímo spjata s řečí, s výroky reálného světa, které jsou pouze nějakým způsobem formalizovány a pravdivostně ohodnoceny. Tento pohled na Aristotelovu logiku je pohledem velmi zjednodušeným (Aristoteles samozřejmě k logice

přistupoval také z filosofické stránky věci), nicméně faktem zůstává, že dvouhodnotová výroková a predikátová logika jsou jeho největším odkazem.[3]

1.2.2 Omyly argumentace v lidské řeči

Nezdá se, že by se argumentace v dnešní společnosti mezi „normálními lidmi“ nějak zvlášť řídila podle zásad Aristoteléské logiky. Přes relativní jednoduchost pravidel dvouhodnotové logiky nejsou základní tautologie pro většinu lidí zřejmé. Jako příklad může posloužit důkaz sporem, který vychází z jednoduché pravdivostní tabulky implikace. Prakticky bez výjimky všichni přijímají tvrzení, že odvození pravdy z pravdivého výroku je logicky správné, zatímco představa, že lze i z nepravdivého výroku odvodit pravdu, je už pro mnoho lidí nepřijatelná. Implikace přitom není nic jiného než tvrzení, že pokud platí premisa, musí platit i závěr. Toto by zřejmě opět připustil každý logicky uvažující člověk, přitom tato formulace nikde nepožaduje, aby se z nepravdy musela odvodit nepravda, tudíž platí, že z nepravdy můžeme klidně odvodit pravdu. V Tabulce 1 lze vidět, že obě dvě formulace jsou ekvivalentní a že tudíž odvození pravdy z nepravdivého předpokladu je zcela správné.[3][4]

Tabulka 1 Pravdivostní tabulka implikace

a	b	$a \Rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Druhým všeobecně rozšířeným nepochopením základních logických spojek je příklad disjunkce, logického „nebo“. Podle pravidel logiky je disjunkce pravdivá, pokud je alespoň jeden výrok pravdivý (viz Tab. 2). Disjunkce je tedy pravdivá i v případě, že jsou pravdivé oba výroky najednou, což často vede k nedorozuměním.[3][4]

Tabulka 2 Pravdivostní tabulka disjunkce

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Prvním problémem je tedy všeobecné nepřijímání některých základních pravidel Aristotelovské logiky. Další nesnáz představují úsudky. Typickým příkladem, který lidem nedělá problémy, je např. následující úsudek: „Pokud venku prší, je mokro. Prší. Závěr: Je mokro.“. V určitých dalších případech je sice ještě možné dosáhnout správného závěru logickou úvahou, ale ta už bude vyžadovat určité zamyšlení, např.: „Lupiči jsou vždy zbabělci, ale ne vždy lháři. Závěr: Lháři nejsou vždy zbabělci.“. Některé (třeba i krátké a jasně formulované) úsudky už je však obtížné, ne-li nemožné, rozhodnout pouhou úvahou, např. (podle L. Carolla): „Nemluvňata nejsou logická. Nikým, kdo nedovede zkrotit krokodýla, nepohrdáme. Kdo je logický, tím pohrdáme. Závěr: Nemluvňata nedovedou zkrotit krokodýla.“. Složitost některých úsudků tedy sama o sobě brání použití zabudované lidské logiky.[3][4]

Dalším problémem je potom charakter dnešního světa, kde lze jen málokdy nalézt čistou pravdu nebo lež. Často není vůbec jasné, jak výrok posoudit, často lze na problém nahlížet relativně, takže se dokonce může stát, že dva lidé stejný výrok ohodnotí opačnou pravdivostní hodnotou. Matematická teorie nabízí řešení ve formě fuzzy logiky, ale tato logika jednak není všemocná (stále musíme umět výrok ohodnotit, i když nyní už do svého rozhodnutí můžeme přidat jakýsi kompromis) a jednak se již skutečně jedná o matematickou disciplínu (i když člověk určitě vnitřně „fuzzy“ hodnocení používá; jedná se však spíše o záležitost související s intuicí než s čistě racionálním myšlením).[2][4]

Vážným problémem, proč se lidé v mnoha situacích nechovají logicky, je z velké části emotivní založení člověka. Ačkoliv se člověk od zvířat odlišuje především svým rozumem, zůstalo mu mnoho emotivních vlastností a pudů. Proto člověk často používá špatnou argumentaci, kterou v některých případech ani argumentací nazvat nelze.[3][4]

Úplně mimo logiku je tzv. argumentace holí („ad baculum“), kde je k přesvědčování používáno úplně jiných prostředků než logického zdůvodňování. Mnohdy používaný způsob, který očividně vychází z citového založení člověka, je používání argumentace společenským nebo individuálním apelem. Snaha argumentovat příslušností k nějaké sociální skupině je velmi často k vidění. Rovněž argumentace individuálním apelem je poměrně častá, jedná se o cílené napadání druhé osoby, např. urážení, obviňování, podezírání apod.[1][3][4]

Oblíbeným způsobem je také dovolávání se vyšší autority. Tento prostředek je velmi účinný a dá se použít v mnoha případech – často je totiž možné dovolávat se i nějaké

nekonkrétní autority, důležité zůstává to, aby oponent autoritu uznal. Proto je možné slyšet „Jeden velmi moudrý muž řekl ...“.[3][4]

Zatím byla zmíněna především jakási „kvaziargumentace“, lidé však často chybují v používání jazyka samotného. Lidská řeč (ať už se jedná o jakýkoliv přirozený jazyk) se bohužel (nebo bohudík) nedá vyjádřit formalizovaným zápisem, proto se v něm zákonitě musí objevovat věci, které je potřeba vyřešit kombinací rozumu a citu. Zde často člověk selhává a mnohdy se uchýlí k některé z následujících chyb: používá víceznačná slova nebo víceznačné věty, chybně skládá úsudky, chybně je rozkládá, pozměňuje význam tvrzení zvýrazněním slov nebo chybnou interpretací slovních tvarů.[1][3][4]

Není těžké najít ještě mnoho dalších chyb, kterých se lidé při zdůvodňování dopouštějí. Na začátku bylo řečeno, že se lidé těchto chyb dopouštějí kvůli svému emotivnímu založení. Dnešní společnost je však k těmto logicky chybným argumentacím částečně rovněž podněcována a někteří jedinci, především politici, ji užívají naprosto záměrně. Politici procházejí specializovaným výcvikem, který bývá vznešeně nazýván např. „výcvik komunikace s médii“ nebo „komunikace s veřejností“. Dostává se jim pokynů, co mají na veřejnosti říkat, jinými slovy jak argumentovat, aby lidi dokázali přesvědčit, že mají pravdu. Pravda jako taková se tedy dostává až do druhořadé role, na prvním místě v jejich snažení je prezentování „své pravdy“ tak, aby byla všeobecně přijata jako pravda absolutní. Politik si totiž podle dnešních „pravidel“ nesmí dovolit přiznat svou chybu, to je něco naprosto nepřijatelného. Proč? Protože přiznáním jakékoliv chyby klesá politikova důvěryhodnost. Lidé chtějí být vedeni pokud možno neomylnými bytostmi, které vždy dělají správné věci, vždy mají správný názor a vždy jsou schopni ho obhájit. A politici to dobře vědí. S každou přiznanou prohrou se snižuje jejich hodnocení, s každým přiznaným selháním nebo lží, chceme-li, klesá jejich šance na znovuzvolení. Musejí tedy, pokud chtějí politiky zůstat i v příštím volebním období, hledat cesty, jak vždy svou pravdu obhájit.[3][4][5]

Chybná argumentace se ukazuje jako ideální prostředek, protože mnoho lidí ji nedokáže odhalit a naopak ji často přijme za uspokojivou a správnou. V zajímavé divadlo se proto mění mnoho televizních politických diskuzí, které jsou spíše než logickým zdůvodňováním zaplněny argumentací účelovou, plnou emotivních výstupů s logicky nepřijatelným vyjadřováním.[4][5]

Tím se ale začarovaný kruh uzavírá. Lidé se dennodenně setkávají se špatnou argumentací, a tak není divu, že ji časem začnou sami používat. Politici jsou krásným vzorem, že „to

jde“. Daří se jim obhajovat až neuvěřitelná stanoviska, nikdy nepřiznají svou chybu, vždy mají v zásobě další „logický“ důvod, proč jednali tak, jak jednali, anebo proč řekli to, co řekli. Proč by se to samé tedy nemělo uplatňovat i při běžném střetu názorů dvou obyčejných lidí? Nikdo nechce přiznat své chyby, každý chce prosadit svou a vypadat přitom důvěryhodně.[4][5]

Logika tudíž dnes mnohdy slouží úplně jinému účelu. Často nejde o to, logicky prokázat, že je něco pravdivé. Naopak, logika je účelově zneužívána. Záměrně jsou hledány konstrukce, které pouze logicky mají vypadat, aniž by byly podle Aristotelovské logiky zdůvodnitelné a obhajitelné. Logika se dokonce stává zajímavou pomůckou nelogické argumentace – toto magické slůvko lze totiž použít jako jeden z individuálních apelů. Výraz „A z toho logicky vyplývá ...“ je velmi silným individuálním apelem, neboť je druhé straně přímo vnučováno, že myslíme přísně logicky a že jakékoliv napadání našeho úsudku je snad něco proti dobrým mravům, něco, co se prostě nedělá. Ze slova „logika“ se tedy stává řečnická vsuvka s vysokou účinností, což je sice smutné, nicméně pravdivé.[4][5]

Dnešní společnost se k logickému odkazu minulosti chová velmi macešsky. Nejen že jsou často nepochopeny některé základní logické zákonitosti (jak bylo ukázáno na příkladu implikace nebo disjunkce), ale někdy je dokonce odkaz logiky zneužíván i v případech, kdy mluvčí ví, že potřebuje argumentovat pomocí jiných pravidel než podle pravidel logiky. Tohoto přístupu je škoda. Hodně by se určitě dalo napravit tím, že by logika byla důkladněji probírána na základní škole, aby každý člověk získal skutečnou představu o tom, co je logicky správné a co nikoliv. Nyní je mnoho lidí odkázáno na způsob, jakým logiku používají jejich rodiče nebo jak slyší argumentovat politiky. Všeobecné rozšíření základních logických znalostí by dnešní společnosti určitě hodně prospělo.[4][5]

2 LOGICKÉ HRY

Hra představuje vedle učení a práce jednu ze tří základních lidských činností. Jedná se o proces aktivní, dynamický, zaměstnávající v menší či větší míře duševní i tělesné schopnosti, které současně cvičí a rozvíjí. Smyslová činnost dítěte bývá nejčastěji motivována prožitky, u dospělých má hra závazná pravidla, cíl nikoli pragmatický, ale ve hře samotné.[6]

Logická hra je tedy duševní činnost člověka prováděná za účelem zábavy či odpočinku, jež rozvíjí logické myšlení.

2.1 Sudoku

Klasické zadání úlohy představuje čtvercová mřížka o devíti řádcích a devíti sloupcích (tedy o 81 prvcích), dále rozdělena na devět nepřekrývajících se souvislých čtvercových bloků – každý o třech řádcích a třech sloupcích (tedy o devíti prvcích), částečně vyplněna čísly (viz Obr. 1). Cíl spočívá v doplnění všech prázdných polí čísly tak, aby v každém řádku, sloupci i bloku bylo každé celé číslo z rozsahu jedna až devět zastoupeno právě jednou (viz Obr. 2).[7]

	1		6					3
				2		1		6
			5	3	1	9		
8			7				1	
	7		1	9	4		8	
	4				3			9
		8	9	1	2			
1		7		5				
9					7		3	

Obrázek 1 Příklad zadání sudoku

2	1	9	6	7	8	5	4	3
3	8	5	4	2	9	1	7	6
7	6	4	5	3	1	9	2	8
8	9	3	7	6	5	4	1	2
6	7	2	1	9	4	3	8	5
5	4	1	2	8	3	7	6	9
4	3	8	9	1	2	6	5	7
1	2	7	3	5	6	8	9	4
9	5	6	8	4	7	2	3	1

Obrázek 2 Řešení zadání sudoku vlevo

2.1.1 Historie

První náznaky logických her s čísly se objevily koncem 19. století ve francouzském tisku. Zadání spočívala v částečně vyplněných „magických čtvercích“, do kterých se měla

doplnit zbývající čísla tak, aby se součet čísel v každém řádku, sloupci i obou úhlopříčkách rovnal určené konstantě.[8]

Za autora prvního moderního sudoku je považován Howard Garns, jehož dílo bylo roku 1979 vydáno v časopise Dell Magazines pod názvem Number Place. V roce 1986 představila v Japonsku tuto hru pod názvem sudoku (v japonštině „Sūji wa dokushin ni kagiru“ znamenající přibližně „čísla jsou jedinečná“) společnost Nikoli. Pod tímto názvem se hra stala roku 2005 celosvětovým fenoménem.[9]

2.1.2 Matematické pozadí

Z definice problému je patrné, že se jedná o speciální případ latinského čtverce, pro který je navíc kladena omezující podmínka týkající se jedinečnosti čísel v každém z bloků.[7]

Řešení klasického sudoku na mřížce tvaru $n^2 \times n^2$ s bloky $n \times n$ představuje NP-úplný problém.[10]

Existuje přibližně $6,7 \times 10^{21}$ různých řešení klasického sudoku. Pokud vyloučíme tvary, které vznikly rotací, zrcadlením nebo permutováním jiného řešení, značně zkrátíme tento počet na zhruba $5,5 \times 10^9$ různých řešení.[11]

Bylo prokázáno, že minimální počet zadaných čísel vedoucí k jednoznačnému řešení je 17.[12]

2.1.3 Metody řešení

Způsobů, jak vyřešit sudoku, existuje celá řada. Všechny ovšem začínají stejně – určením tzv. „kandidátů“ pro každou pozici. Kandidáti představují množinu všech přípustných čísel dané buňky, které se zatím nepodařilo explicitně vyloučit, a tak kterékoliv z nich může být řešením této pozice.[7]

Na začátku (ještě před použitím jakýchkoliv řešících technik či úvah) bude v každém prázdném poli všech devět kandidátů (viz Obr. 3). Pro lepší přehlednost je vhodné zapisovat kandidáty o stejných hodnotách na stejná místa.[7]

<small>1 2 3 4 5 6 7 8 9</small>	1	<small>1 2 3 4 5 6 7 8 9</small>	6	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	3
<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	2	<small>1 2 3 4 5 6 7 8 9</small>	1	<small>1 2 3 4 5 6 7 8 9</small>	6
<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	5	3	1	9	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>
8	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	7	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	1	<small>1 2 3 4 5 6 7 8 9</small>
<small>1 2 3 4 5 6 7 8 9</small>	7	<small>1 2 3 4 5 6 7 8 9</small>	1	9	4	<small>1 2 3 4 5 6 7 8 9</small>	8	<small>1 2 3 4 5 6 7 8 9</small>
<small>1 2 3 4 5 6 7 8 9</small>	4	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	3	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	9
<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	8	9	1	2	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>
1	<small>1 2 3 4 5 6 7 8 9</small>	7	<small>1 2 3 4 5 6 7 8 9</small>	5	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>
9	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	<small>1 2 3 4 5 6 7 8 9</small>	7	<small>1 2 3 4 5 6 7 8 9</small>	3	<small>1 2 3 4 5 6 7 8 9</small>

Obrázek 3 Úvodní výpis všech kandidátů

V prvním kroku jsou odstraněni kandidáti z volných polí podle základních pravidel. Pro každé zadané číslo jsou smazáni kandidáti o této hodnotě z příslušného řádku, sloupce i bloku (viz Obr. 4).[7]

<small>2 4 5 7</small>	1	<small>2 4 5 9</small>	6	<small>4 7 8</small>	<small>8 9</small>	<small>2 4 5 7 8</small>	<small>2 4 5 7</small>	3
<small>3 4 5 7</small>	<small>3 5 8 9</small>	<small>3 4 5 9</small>	<small>4 8</small>	2	<small>8 9</small>	1	<small>4 5 7</small>	6
<small>2 4 6 7</small>	<small>2 6 8</small>	<small>2 4 6</small>	5	3	1	9	<small>2 4 7 8</small>	<small>2</small>
8	<small>2 3 5 6 9</small>	<small>2 3 5 6 9</small>	7	<small>6</small>	<small>5 6</small>	<small>2 3 4 5 6</small>	1	<small>2 4 5</small>
<small>2 3 5 6</small>	7	<small>2 3 5 6</small>	1	9	4	<small>2 3 5 6</small>	8	<small>2 5</small>
<small>2 5 6</small>	4	<small>1 2 5 6</small>	<small>2 8</small>	<small>6 8</small>	3	<small>2 5 6 7</small>	<small>2 5 6</small>	9
<small>3 4 5 6</small>	<small>3 5 6</small>	8	9	1	2	<small>4 5 6 7</small>	<small>4 5 6 7</small>	<small>4 5 7</small>
1	<small>2 3 6</small>	7	<small>4 3 8</small>	5	<small>6 8</small>	<small>2 4 6 8</small>	<small>2 4 6 9</small>	<small>4 8</small>
9	<small>2 5 6</small>	<small>2 4 5 6</small>	<small>4 8</small>	<small>4 6 8</small>	7	<small>2 4 5 6 8</small>	3	<small>1 2 4 5 8</small>

Obrázek 4 Kandidáti po aplikaci elementárních pravidel

V tomto bodě se řešící techniky rozbíhají. Jakýkoliv následný postup řešení je ovšem vždy posloupností kroků skládajících se z opakovaně používaných metod řešení (od nejjednodušších až po velmi složité). Obecně se jedná o rozpoznání určitého vzoru (uspořádání buněk, hodnot, kandidátů nebo vazeb mezi nimi) a uplatnění odpovídající akce. Konkrétní podoba akce pak závisí na podmínkách jejího vzniku – např. hodnota vyřešené pozice je závislá na hodnotách a kandidátech v buňkách, kterých se akce týká.[7]

Akce lze typově rozdělit do tří hlavních kategorií.[7]

1. Doplnění konečné hodnoty do buňky, tj. vyřešení pozice, pokud zbývá jediná možnost. Obecně je těchto pravidel velmi málo. Např. „Naked Singles“ nebo „Hidden Singles“.[7]
2. Odstranění kandidátů. Většina pravidel je tohoto typu. Obecně vyjadřují, že za daných okolností se určitá čísla nemohou vyskytovat na daných pozicích, proto je možné tyto kandidáty z buněk odstranit. Např. aplikace elementárních pravidel na začátku a automaticky po každém doplnění výsledné hodnoty, tj. odstranění kandidátů o hodnotě doplněného čísla z příslušného řádku, sloupce a bloku.[7]
3. Metoda pokusu a omylu. Jedná se o rekurzivní tipování výsledné hodnoty dané pozice ze zbývajících kandidátů, zkoumání následků a případné ponechání této hodnoty či odstranění odpovídajícího kandidáta. V praxi se tato metoda využívá zřídka. Důvodem je jednak velká časová náročnost pro člověka i dnešní počítače a jednak značná monotónnost a nezábavnost. Při použití pokročilejších technik je jen málokdy dosaženo situace, kdy by bylo nutné tuto metodu využít.[7]

Vzhledem k rozsahu a zaměření této práce si dovoluji uvést charakteristiky pouze vybraných metod řešení, jejichž kombinací bude programově možné velmi rychle vyřešit jakékoliv zadání sudoku nebo jej naopak vygenerovat.

Pokročilých technik pro řešení člověkem existuje celá řada, kdyby tomu tak nebylo a stačily by základní postupy, pravděpodobně by se sudoku nikdy neproslavilo.[7]

2.1.3.1 Naked Singles (NS, zkráceně také Singles, S)

Pokud pro buňku zbývá jediný kandidát, je tato hodnota řešením pozice. Vzorový příklad doplnění čísla 6 zmíněnou technikou do prostředního bloku znázorňují Obrázky 5 a 6.[7]

2 4 5 7	1	2 4 5 9	6	4 7 8	8 9	2 4 5 7 8	2 4 5 7	3
3 4 5 7	5 8 9	3 4 5 9	4 8	2	8 9	1	4 5 7	6
2 4 6 7	2 6 4 6	2 6 4 6	5	3	1	9	2 4 7	2 4 7 8
8	2 3 5 6 9	2 3 5 6 9	7	6	5 6	2 3 4 5 6	1	2 4 5
2 3 5 6	7	2 3 5 6	1	9	4	2 3 5 6	8	2 5
2 5 6	4	1 2 5 6	2 8	6 8	3	2 5 6 7	2 5 6 7	9
3 4 5 6	3 5 6	8	9	1	2	4 5 6 7	4 5 6 7	4 5 7
1	2 3 6	7	4 8	3 5	6 8	2 4 6 8	2 4 6 9	2 4 8
9	2 5 6	2 4 5 6	4 8	4 6 8	7	2 4 5 6 8	3	1 2 4 5 8

Obrázek 5 Příklad „Naked Single“
v prostředním bloku

2 4 5 7	1	2 4 5 9	6	4 7 8	8 9	2 4 5 7 8	2 4 5 7	3
3 4 5 7	5 8 9	3 4 5 9	4 8	2	8 9	1	4 5 7	6
2 4 6 7	2 6 4 6	2 6 4 6	5	3	1	9	2 4 7	2 4 7 8
8	2 3 5 6 9	2 3 5 6 9	7	6	5	2 3 4 5	1	2 4 5
2 3 5 6	7	2 3 5 6	1	9	4	2 3 5 6	8	2 5
2 5 6	4	1 2 5 6	2 8	6 8	3	2 5 6 7	2 5 6 7	9
3 4 5 6	3 5 6	8	9	1	2	4 5 6 7	4 5 6 7	4 5 7
1	2 3 6	7	4 8	3 5	6 8	2 4 6 8	2 4 6 9	2 4 8
9	2 5 6	2 4 5 6	4 8	4 6 8	7	2 4 5 6 8	3	1 2 4 5 8

Obrázek 6 Doplnění „Naked Single“
z obrázku vlevo

2.1.3.2 Hidden Singles (HS)

Jestliže je kandidát o určité hodnotě jedinečný v rámci svého řádku, sloupce nebo bloku, musí být řešením své buňky. Ukázka doplnění hodnoty 1 prostřednictvím této metody do levého prostředního bloku je znázorněna na Obrázcích 7 a 8.[7]

2 4 5 7	1	2 4 5 9	6	4 7 8	8 9	2 4 5 7 8	2 4 5 7	3
3 4 5 7	5 8 9	3 4 5 9	4 8	2	8 9	1	4 5 7	6
2 4 6 7	2 6 4 6	2 6 4 6	5	3	1	9	2 4 7	2 4 7 8
8	2 3 5 6 9	2 3 5 6 9	7	6	5 6	2 3 4 5 6	1	2 4 5
2 3 5 6	7	2 3 5 6	1	9	4	2 3 5 6	8	2 5
2 5 6	4	1 2 5 6	2 8	6 8	3	2 5 6 7	2 5 6 7	9
3 4 5 6	3 5 6	8	9	1	2	4 5 6 7	4 5 6 7	4 5 7
1	2 3 6	7	4 8	3 5	6 8	2 4 6 8	2 4 6 9	2 4 8
9	2 5 6	2 4 5 6	4 8	4 6 8	7	2 4 5 6 8	3	1 2 4 5 8

Obrázek 7 Příklad „Hidden Single“
v levém prostředním bloku

2 4 5 7	1	2 4 5 9	6	4 7 8	8 9	2 4 5 7 8	2 4 5 7	3
3 4 5 7	5 8 9	3 4 5 9	4 8	2	8 9	1	4 5 7	6
2 4 6 7	2 6 4 6	2 6 4 6	5	3	1	9	2 4 7	2 4 7 8
8	2 3 5 6 9	2 3 5 6 9	7	6	5 6	2 3 4 5 6	1	2 4 5
2 3 5 6	7	2 3 5 6	1	9	4	2 3 5 6	8	2 5
2 5 6	4	1 2 5 6	2 8	6 8	3	2 5 6 7	2 5 6 7	9
3 4 5 6	3 5 6	8	9	1	2	4 5 6 7	4 5 6 7	4 5 7
1	2 3 6	7	4 8	3 5	6 8	2 4 6 8	2 4 6 9	2 4 8
9	2 5 6	2 4 5 6	4 8	4 6 8	7	2 4 5 6 8	3	1 2 4 5 8

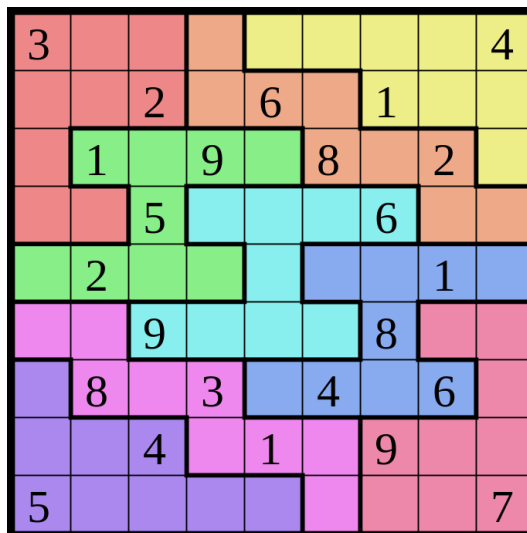
Obrázek 8 Doplnění „Hidden Single“
z obrázku vlevo

2.1.3.3 Backtracking (BT, česky nejčastěji metoda zpětného vyhledávání)

Viz „metoda pokusu a omylu“ na straně 19.

2.1.4 Varianty

Vedle výše zmíněné zdaleka nejrozšířenější verze existuje nepřeborné množství jejích mutací, počínaje od změny rozměrů hracího pole (které vůbec nemusí být čtvercové) přes nahrazení čísel za písmena či jiné speciální znaky až po vytvoření jakýchsi více či méně pravidelných útvarů uvnitř původní mřížky, ve kterých musí být rovněž každé číslo jedinečné (viz Obr. 9).



Obrázek 9 Příklad zadání sudoku
s nepravidelnými bloky[13]

2.2 Zebra puzzle

Zebra puzzle jsou zobecněním původní Einsteinovy hádanky. Název je odvozen z hlavního úkolu – zjistit který z majitelů domů chová zebra.

Znění původní hádanky údajně vyřkl mladý Albert Einstein, odtud její pojmenování. Spolu se zadáním prý také odhadoval, že 98 % populace ji nebude schopno rozluštit.[14]

Einsteinova hádanka v sobě neskrývá žádná tajemství ani triky, a je tak považována za jeden z nejtěžších čistě logických testů, které kdy byly vymyšleny.[14]

2.2.1 Originální text hlavolamu

1. *There are five houses.*
2. *The Englishman lives in the red house.*
3. *The Spaniard owns the dog.*
4. *Coffee is drunk in the green house.*

5. *The Ukrainian drinks tea.*
6. *The green house is immediately to the right of the ivory house.*
7. *The Old Gold smoker owns snails.*
8. *Kools are smoked in the yellow house.*
9. *Milk is drunk in the middle house.*
10. *The Norwegian lives in the first house.*
11. *The man who smokes Chesterfields lives in the house next to the man with the fox.*
12. *Kools are smoked in the house next to the house where the horse is kept.*
13. *The Lucky Strike smoker drinks orange juice.*
14. *The Japanese smokes Parliaments.*
15. *The Norwegian lives next to the blue house.*

Now, who drinks water? Who owns the zebra?

In the interest of clarity, it must be added that each of the five houses is painted a different color, and their inhabitants are of different national extractions, own different pets, drink different beverages and smoke different brands of American cigarets. One other thing: in statement 6, right means your right.[15]

2.2.2 Varianty

První verze byla roku 1962 otištěna v časopise Life International. Od té doby vzniklo nespočet jejích obměn. Nejběžnější varianty originální verze s pěti domy a pěti jejich vlastnostmi představuje nahrazování hodnot jednotlivých vlastností (např. místo psa chová Španěl kanárka) nebo záměna atributu jako celku (zejména nahrazování cigaretových značek např. za sportovní odvětví, aby byli čtenáři podvědomě vedeni ke zdravějšímu životnímu stylu). Záměny tohoto druhu mají na řešení samozřejmě pouze kosmetický dopad.

Hlubší vliv na řešení už bude mít jiná sada vodítek, tj. provázání odlišných vlastností nebo použití jiných relací (např. místo „Angličan žije v červeném domě.“ bude zadáno „Angličan nežije v modrém domě.“).

Dále je možné nezávisle na sobě měnit počet domů a počet vlastností, takže klidně může existovat zadání se dvěma domy a čtyřmi vlastnostmi.

Zadání by vždy mělo vést k jednoznačnému řešení a dobrým zvykem je také minimální počet vodítek, což znamená, že odstraněním kteréhokoliv z nich dojde ke ztrátě oné jedinečnosti řešení.

Na pořadí vodítek ani vlastností nezáleží, tj. lze je libovolně permutovat.

2.2.3 Metody řešení

Klíčem k rozluštění hádanek tohoto typu je vytvoření přehledné tabulky (viz Tab. 3). Počet řádků je dán počtem vlastností každého domu (např. národnost majitele, barva domu, domácí mazlíček, ...). Každý sloupec pak bude obsahovat vlastnosti jednoho domu.[14]

Tabulka 3 Výchozí stav klasické Einsteinovy hádanky

Dům	1	2	3	4	5
Národnost					
Barva					
Mazlíček					
Nápoj					
Cigarety					

Cíl spočívá ve vyplnění všech prvků tabulky, takže ať se zadavatel úkolu zeptá na cokoliv, bude možno odpověď na jeho otázku v tabulce nalézt.

Pokud číslování domů není zadáno přímo, lze si zvolit buď číslování zleva doprava, anebo zprava doleva, tedy jestli dům číslo jedna bude nejvíce vlevo, nebo vpravo. Tak či onak bude dosaženo stejného výsledku (samozřejmě vyjma čísel domů).

Nejlepší je začít vyhledáním mezi vodítky taková, která se vztahují k číslům domů (např. „Nor žije v prvním domě“), protože tyto informace je možné zanést do tabulky přímo. Využití nápovědy je pro lepší orientaci v zadání vhodné škrtnat. Dále je postupováno cestou přímého doplňování pomocí vodítek, které souvisí s již vyplněnými údaji (např. z „Nor žije vedle modrého domu.“ je možné vyvodit, že dům číslo dva je modrý, protože první dům jiného souseda nemá). U většiny hlavolamů je však dosaženo stádia, kdy touto přímou cestou není možné dojít až do cíle a je třeba zvolit pokročilejší postup. Na výběr je více možností.[14]

2.2.3.1 Metoda hypotéz

Méně vznešeným názvem také „metoda pokusu a omylu“. Vybere se buňka tabulky, na kterou aspiruje co nejméně kandidátů (nejlépe dva), protože zde je největší šance se trefit

(např. nelze přímo určit, zda majitel třetího domu chová lišku nebo koně). Zkusí se doplnit jedna z možností a uvážením zbývajících vodítek je určeno, jestli tato možnost do řešení zapadá. Pokud je dosaženo rozporu s ostatními nápovědami (např. domu se zatím neurčenou barvou nelze přiřadit žádná ze zbývajících), je tato možnost zavržena a zkusí se jiná. V případě tipování ze dvou možností je zřejmé, že druhá musí být správná.[16]

Touto cestou lze vždy zaručeně dojít až do cíle, avšak může to trvat dlouho a v případě „košatého“ stromu možností se v něm lze snadno zamotat, proto je tato technika doporučována až jako poslední možnost.

2.2.3.2 Metoda párování

Některé zatím nepřirazené vlastnosti lze dle zadání spárovat (např. „Ukrajinec pije čaj.“ tvoří pár „Ukrajinec – čaj“), zatímco několik zůstane nepárových. Je zřejmé, že např. domu, kterému chybí doplnit tři vlastnosti, mohou tyto být buď tři nepárové, anebo jeden pár a jedna nepárová. Lze vyloučit párové i nepárové vlastnosti, které již jsou u tohoto domu určeny. Zbyde-li např. pouze jediná nepárová vlastnost, jistě bude tomuto domu náležet, neboť obecně při lichém počtu nevyplněných vlastností je nutno vždy využít alespoň jednu nepárovou vlastnost.[16]

Předností této techniky je rychlost, proto je vhodná hned po přímé cestě. Úspěch ovšem není zaručen.

2.2.3.3 Vylučovací metoda

Při označení počtu vlastností r a počtu domů c , vytvoříme namísto Tabulky 3 o rozměrech $r \times c$ zmiňované výše, Tabulku 4 o rozměrech $r * c \times c$.[16]

Tabulka 4 Výchozí stav zadání se třemi domy a dvěma vlastnostmi

Dům	1	2	3
Národnost	Čech	Čech	Čech
	Rus	Rus	Rus
	Američan	Američan	Američan
Barva	Bílá	Bílá	Bílá
	Červená	Červená	Červená
	Modrá	Modrá	Modrá

Vodítka jsou cyklicky procházena a z Tabulky 4 jsou vyškrťovány nepřipustné možnosti.[16]

Např. uplatněním vodítka „Čech bydlí ve druhém domě.“ se Tabulka 4 změnila na Tabulku 5.

Tabulka 5 Předchozí tabulka po aplikaci zmíněného vodítka

Dům	1	2	3
Národnost		Čech	
	Rus		Rus
	Američan		Američan
Barva	Bílá	Bílá	Bílá
	Červená	Červená	Červená
	Modrá	Modrá	Modrá

Výhodou této metody je její jednoduchost a nenáročnost na přemýšlení. Mezi nedostatky patří monotónnost, nezábavnost a především velká časová náročnost, proto její použití u testů SCIO či LSAT není doporučeno.

2.2.3.4 Srovnání metod

Neexistuje nejlepší metoda, každá má své výhody i nevýhody. Záleží na typu člověka i situaci, každému sedí něco jiného a pro každou situaci se hodí něco jiného. Rozhodně není na škodu umět metod více a kombinovat je nebo se podle zkušeností rozhodnout, jakou techniku na danou situaci nasadit. Mně osobně nejvíce vyhovuje kombinace metody párování a hypotéz.

2.3 Puzzle

Puzzle (skládanka) představuje soubor různě tvarovaných dílků, z nichž jen některé do sebe zapadají. Každý dílek skýtá část výsledného obrázku (mozaiky), který vznikne jejich správnou kombinací.

Mezi nejčastější motivy puzzle patří příroda (zejm. pohoří) a budovy (zejm. hrady). Některé společnosti dokonce nabízí vytvoření skládanky na zakázku podle dodané fotografie.

Největšího hráče na evropském trhu dlouhodobě představuje německá firma Ravensburger. Skládání puzzle rozvíjí kreativitu, logické myšlení a odbourává stres, čímž snižuje riziko výskytu Alzheimerovy choroby.[17]

2.3.1 Historie

Za prvního autora puzzle je považován londýnský kartograf John Spilsbury, který kolem roku 1760 připevnil jednu ze svých map na dřevo a rozřezal podél státních hranic. Výtvar sloužil britským školákům, kteří se na něm zábavnou formou učili zeměpis. Nápad se uchytil pod názvem „dissections“ (pitvy) a přibližně do roku 1820 sloužil výhradně vzdělávacím účelům. Kolem roku 1880 začal převládat název „jigsaw puzzle“ (skládačka vyřezávaná pilou). Koncem 19. století se pro výrobu kvalitnějších skládanek začala používat překližka, zatímco ty levnější byly vyráběny z lepenky. Na jednu stranu se přilepil nebo namaloval obrázek a na druhou se tužkou vyznačila místa řezu. Tahy tužkou lze ještě dnes nalézt na zadních stranách některých starších puzzle.[18]

Vrcholu své popularity dosáhly puzzle ve 20. a 30. letech 20. století, kdy nejčastěji zobrazovaly romantické výjevy nebo nadšení z technologického pokroku v oblasti lodní a železniční dopravy. Na první pohled se může zdát zvláštní, že se právě puzzle, které nejsou k životu potřebné, velmi dobře prodávaly během ekonomické krize. Důvodem bylo, že si člověk za nízkou cenu (do jednoho amerického dolaru) zajistil zábavu na několik hodin. Hrát bylo možno buď samostatně, anebo ve skupině. Navíc bylo možné skládanou po dokončení rozložit a vyměnit s kamarádem za jinou.[18]

Od 40. let až po současnost se obliba puzzle potýká s úpadkem. Sice se někde stále používají jako učební pomůcka v zeměpisu, pořád se vyrábějí z překližky a lepenky a stále platí, že za málo peněz koupí člověk hodně zábavy, ale bohužel jsou v dnešní době vytlačovány nezdravou, hloupou, vulgární a násilnou televizní či počítačovou zábavou.[18]

2.3.2 Varianty

Na obtížnost má největší vliv počet dílků, který se pohybuje od několika kusů až po tisíce.

V poslední době se na trhu lze běžně setkat s trojrozměrnými puzzle – buď jsou 3D už samotné dílky (viz Obr. 10; jedná se o tzv. „puzzle ball“, který nejčastěji znázorňuje moderní či historický glóbus, ale má ho ve svém logu také např. Wikipedie), anebo tyto zůstávají dvourozměrné a skládají se postupně ve vrstvách na sebe (viz Obr. 11).

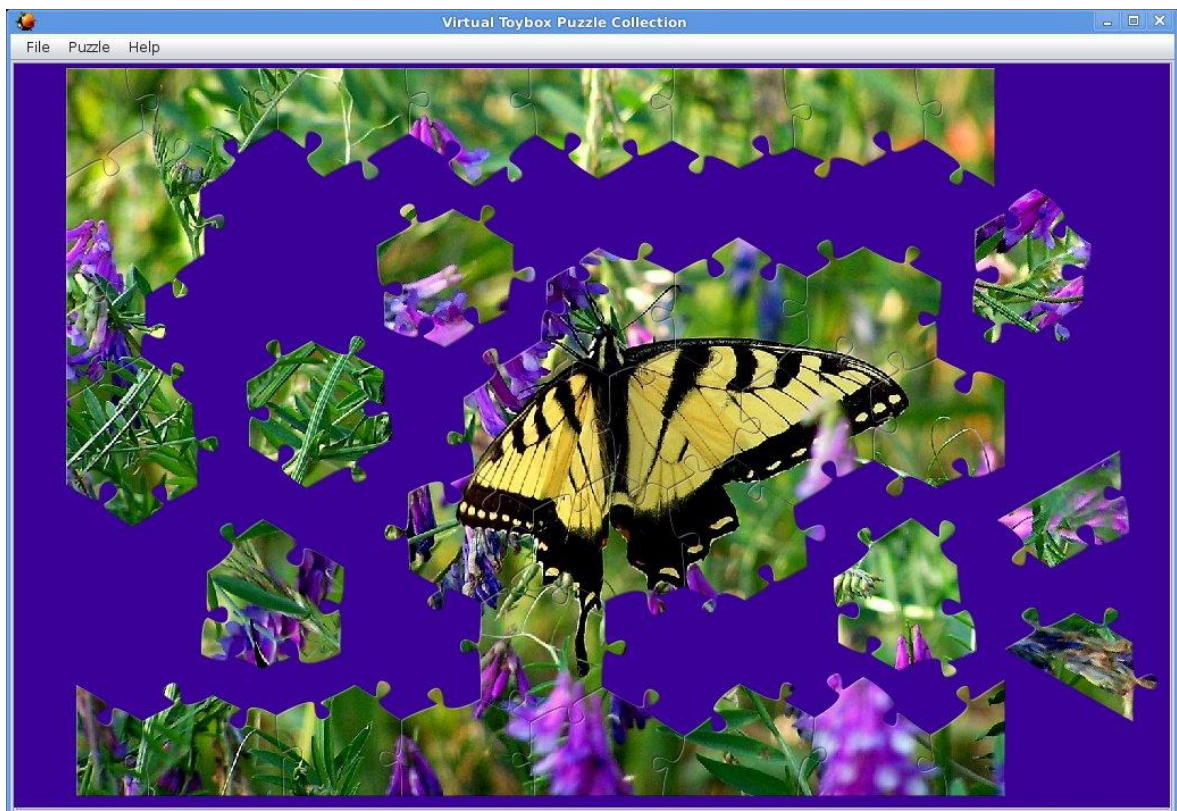


Obrázek 10 Logo Wikipedie[19]



Obrázek 11 Vrstvené trojrozměrné puzzle[20]

Existují také počítačové verze, které jsou zdarma, nevyžadují úklid a není třeba se obávat, že se dílek ztratí nebo poničí (viz Obr. 12).



Obrázek 12 Software ke stažení zdarma pro skládání puzzle umožňující rotaci dílků[21]

2.3.3 Světové rekordy

Největší sériově vyráběnou skládanku prodává od roku 2010 německá společnost Ravensburger. Obsahuje 32 256 dílků, které znázorňují 32 děl amerického umělce Keitha Haringa a po složení tvoří obdélník o stranách 544 cm a 192 cm.[22]

Světově největší puzzle zapsané v Guinnessově knize rekordů byly v roce 2002 složeny 777 lidmi na letišti v Hong Kongu. Skládanku představovalo 21 600 čtvercových dílků, každý o straně 50 cm, což vytvořilo obrazec na ploše větší než 5 400 m². [23]

3 MOŽNOSTI WEBOVÉ IMPLEMENTACE

Existují dva způsoby, jak je možné zpřístupnit aplikaci napsanou v programu Mathematica na webu. Prvním z nich je využití nástavby webMathematica, druhým uložení kódu ve formátu CDF (Computable Document Format).[24]

3.1 webMathematica

System webMathematica pracuje na bázi klient-server. Klient odešle kód napsaný v SW Mathematica na server, který jej vyhodnotí a pošle zpět na klienta výsledek.[24]

Část kódu v programu webMathematica obsluhující události od uživatele (tzn. vše založené na funkci *Dynamic*) nemůže být napsána pomocí kódu Mathematica, protože toto není na straně serveru podporováno. Uživatelské rozhraní tedy musí být vytvořeno pomocí HTML, dynamičnost je možno webové stránce dodat například využitím technologií JavaScript nebo Flash.[24]

Aplikace poběží na každém klientu, který má nainstalován moderní webový prohlížeč. Pouze aplikace využívající funkci *MSPManipulate* ke svému chodu vyžadují ještě Flash plugin.[24]

Veškerou vizualizaci výsledků zajišťuje server a následně posílá na klienta, což může zejména u větších souborů dat značně omezovat plynulou interakci.[24]

3.2 CDF

V případě kódu Mathematica obsaženém v dokumentu CDF se žádná data na server neodesílají, vše zpracovává klient – ať už se jedná o interpretaci kódu v samostatném CDF přehrávači nebo v zásuvném modulu (pluginu) webového prohlížeče.[24]

Dynamické uživatelské rozhraní musí využívat funkci *Dynamic* programu Mathematica.[24]

Spuštění CDF dokumentu je možné buď na plné verzi SW Mathematica nebo na samostatném CDF přehrávači (FreeCDF, EnterpriseCDF nebo Player Pro). Tyto SW však samozřejmě podporuje mnohem méně zařízení oproti HTML prohlížečům.[24]

3.3 Srovnání

Uvážením rozdílů mezi oběma technologiemi lze zjistit, že spolu nesoupeří, ale doplňují se. Je tedy možné a u větších projektů také výhodné obě varianty kombinovat. Například

vytvořit aplikaci napsanou v SW webMathematica, která vrací CDF dokument umožňující uživateli dynamickou interakci s výsledkem, který byl spočítán na serveru.[24]

Vzhledem k vysokým požadavkům her na dynamičnost tedy implementace pomocí systému webMathematica není možná a vytvoření kombinovaného projektu přesahuje rámec této bakalářské práce.

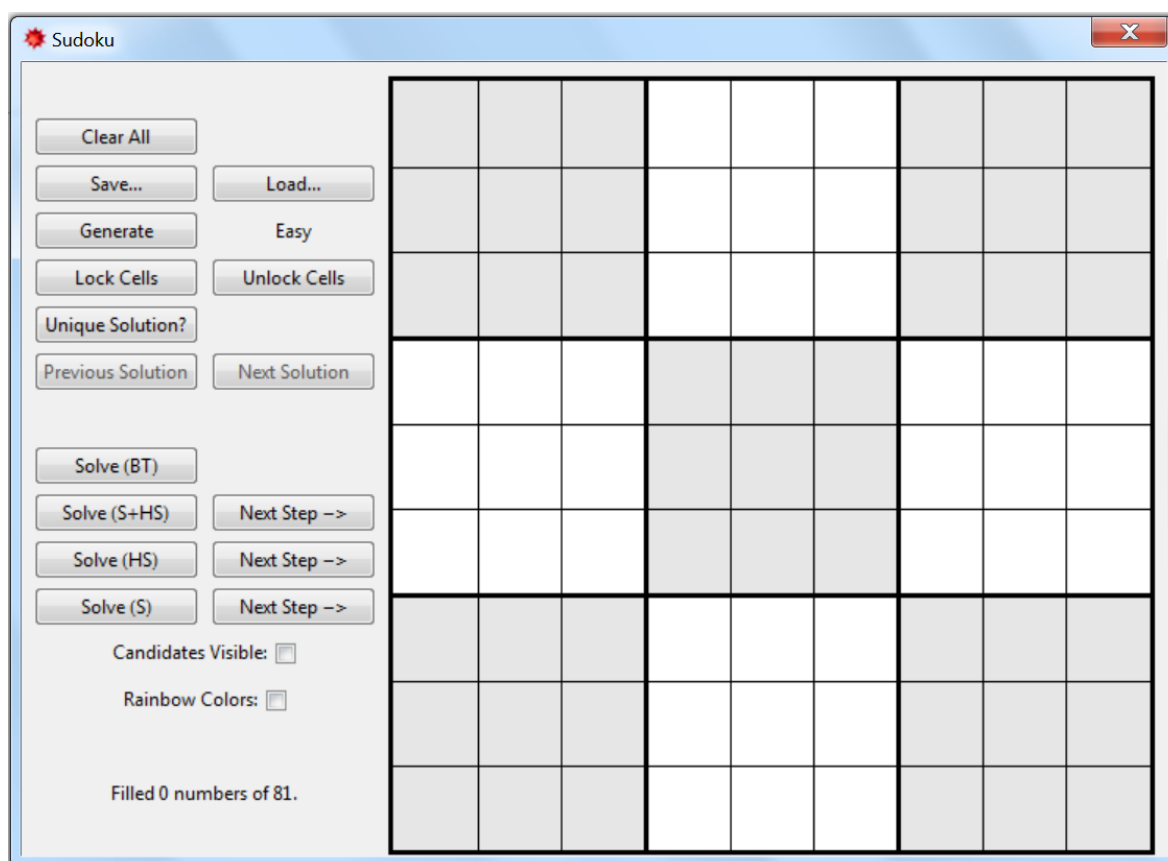
II. PRAKTICKÁ ČÁST

4 SUDOKU

Počítačová verze zdaleka nejrozšířenější varianty 9×9 hry sudoku nabízí intuitivní uživatelské rozhraní s funkcemi pro vytváření zadání různých obtížností, jejich vyřešení odlišnými technikami, testem jedinečnosti řešení, možností nápovědy a uložení či načtení rozehrané hry ze souboru.

4.1 Uživatelský manuál

Celá aplikace funguje přehledně v jediném plně interaktivním dialogovém okně. Následující Obrázek 13 uživatel uvidí na svém počítači bezprostředně po spuštění programu.



Obrázek 13 Výchozí stav dialogového okna po spuštění programu Sudoku

V levé části lze nalézt tlačítka, zaškrtnutí políčka, rozevřací seznam a nejrůznější informační hlášky, napravo pak sudoku mřížku.

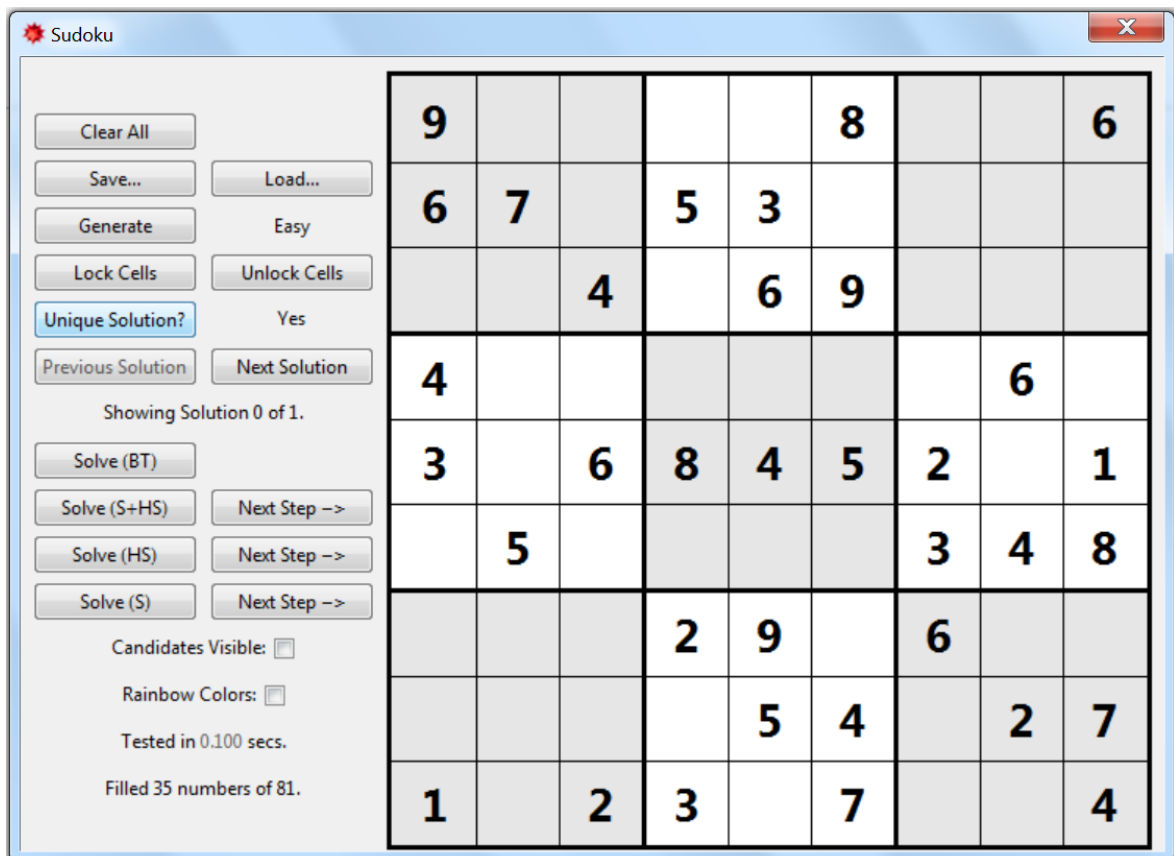
4.1.1 Mřížka

Po najetí kurzorem myši přibližně do středu každého pole mřížky se objeví rozevřací seznam s možností zadání či smazání čísla. Hodnoty zadané uživatelem jsou modré, čísla

vyplněná počítačem nebo tzv. „zamknutá“ tlačítkem *Lock Cells* jsou černá a needitovatelná. Pole je možné „odemknout“ tlačítkem *Unlock Cells*. V případě ponechání „automatické kontroly“ výskytu dvou a více stejných hodnot v rámci jednoho řádku, sloupce či bloku (zaškrťovací políčko *Rainbow Colors* je prázdné) jsou tyto označeny červeně.

4.1.2 Tlačítka

- *Clear All* – Plnohodnotný návrat do výchozího stavu.
- *Save...* – Otevře nativní dialogové okno příslušného operačního systému pro uložení souboru. Výchozím adresářem je složka, ze které byla aplikace spuštěna. Do souboru CSV ukládá pouze hodnoty v mřížce. Oddělovačem je vždy středník. Vytvořený soubor zabírá jen 169 bytů paměti a je upravovatelný v jakémkoliv textovém editoru nebo např. programu MS Excel.
- *Load...* – Otevře nativní dialogové okno příslušného operačního systému pro načtení souboru. Výchozím adresářem je složka, ze které byla aplikace spuštěna. Ze souboru CSV načítá pouze hodnoty do mřížky. Oddělovačem může být čárka nebo středník (dokonce lze použít kombinaci obou v rámci jednoho souboru).
- *Generate* – Vytváří zadání s jedinečným řešením a ohledem na obtížnost zvolenou z rozevíracího seznamu na témž řádku. Čím vyšší obtížnost, tím delší dobu výpočet trvá. V závislosti na HW a „štěstí“, na kolikátou iteraci se podaří nalézt zadání splňující všechny požadavky, se celková doba pohybuje od několika málo sekund pro nejjednodušší obtížnost až po několik desítek vteřin pro tu nejtěžší. Maximální čas, po který se program snaží vytvořit zadání a „neodpovídá“, je stanoven na 60 sekund.
- *Lock Cells* – Znemožní další editování hodnot v mřížce, které jsou již vyplněny.
- *Unlock Cells* – Umožní úpravu všech polí mřížky.
- *Unique Solution?* – Hledá dvě různá řešení pro aktuální soubor hodnot v mřížce, přičemž nalezená řešení si ukládá do paměti a dle jejich počtu rozhoduje, zda vůbec nějaké řešení existuje, a pokud ano, jestli je jedinečné (viz Obr. 14). Případné nalezené řešení automaticky nezobrazuje, ale čeká až na stisknutí tlačítka *Next Solution*. Maximální čas, po který se program snaží ověřit jedinečnost řešení a „neodpovídá“, je stanoven na 60 sekund.

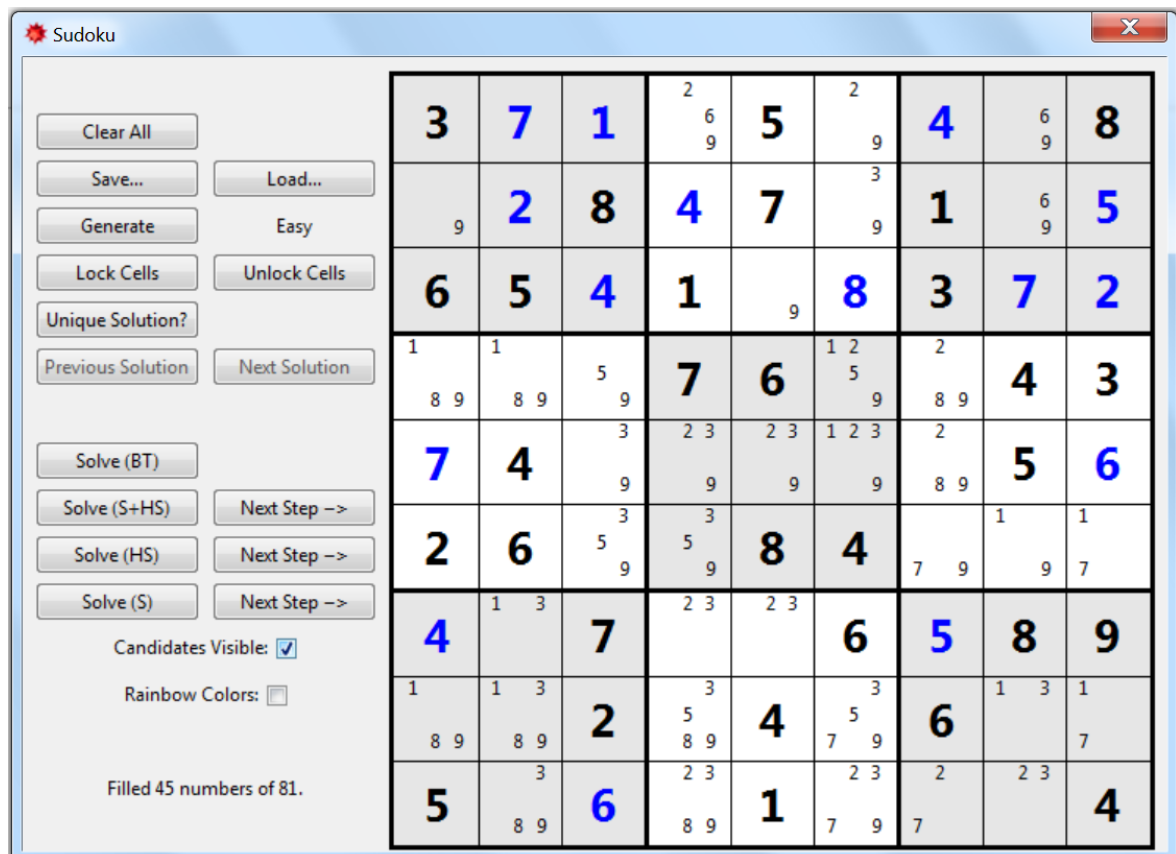


Obrázek 14 Ověření jedinečnosti řešení pro zadání v mřížce

- *Previous Solution* – Listuje souborem nalezených řešení vzad.
- *Next Solution* – Listuje souborem nalezených řešení vpřed.
- *Solve* – Pomocí různých metod hledá jedno řešení pro aktuální soubor hodnot v mřížce.
 - *BT* – Využívá metod *singles*, *hidden singles* a *backtracking*.
 - *S+HS* – Využívá metod *singles* a *hidden singles*.
 - *HS* – Využívá metody *hidden singles*.
 - *S* – Využívá metody *singles*.
- *Next Step* – U odpovídajících metod vykoná pouze jeden krok algoritmu.

4.1.3 Zaškrťovací políčka

- *Candidates Visible* – V nevyplněných polích mřížky zobrazuje zbývající kandidáty (viz Obr. 15).



Obrázek 15 Nápopověda zbývajících kandidátů u nevyplněných polí

- *Rainbow Colors* – Každému číslu přidělí jedinečnou barvu, což zlepšuje přehlednost, avšak vypíná „automatickou kontrolu“ a „zamknutá“ pole nejsou barevně odlišena.

4.1.4 Rozevírací seznam

Při najetí myši na text napravo od tlačítka *Generate* se objeví rozevírací seznam s položkami pro volbu obtížnosti generovaného zadání.

- *Easy* – Lze vyřešit pouze metodou *singles*.
- *Medium* – Nelze vyřešit pouze metodou *singles*, ale lze vyřešit pouze metodou *hidden singles*.
- *Hard* – Nelze vyřešit pouze metodou *singles* ani pouze metodou *hidden singles*, ale lze vyřešit kombinací metod *singles* a *hidden singles*.
- *Expert* – Nelze vyřešit ani kombinací metod *singles* a *hidden singles*, ale lze vyřešit některou z pokročilejších metod.

4.2 Programátorský manuál

Následující podkapitoly nastiňují hlavní myšlenky a principy klíčových částí vnitřní struktury programu *Sudoku*.

4.2.1 Deterministické metody řešení

Hlavní funkce pro deterministické určování zbývajících hodnot v mřížce *candidatesSolutions* využívá řešících metod *singles* a *hidden singles*. Vstupními parametry jsou *values* – zadání mřížky, *oneStep* – má-li se vykonat pouze jeden krok algoritmu, *singles* – má-li se použít metoda *singles* a *hiddenSingles* – má-li se použít metoda *hiddenSingles*.

Inicializuje se 3D pole *candidates*, ve kterém první dva rozměry představují řádky a sloupce mřížky, zatímco třetí rozměr tvoří kandidáti na danou pozici.

Do *candidates* jsou zavedeny zadané hodnoty a pomocí funkce *removeCandidates* odstranění odpovídající kandidáti z ostatních buněk. Dále jsou ve smyčce *while* cyklicky aplikovány povolené řešící algoritmy a vždy po doplnění nového čísla volána funkce *removeCandidates* a nastaven příznak *true* do proměnné *nextStep*, aby program věděl, zda má ve smyčce *while* setrvávat a řešit dále, anebo cyklus ukončit a vrátit aktuální řešení – ať už úplné či neúplné. V případě volání funkce s hodnotou *true* v parametru *oneStep* se smyčka *while* prochází vždy právě jednou.

4.2.2 Stochastické metody řešení

Funkce *backtrackSolutions* využívající algoritmu *backtracking* slouží pro hledání řešení nedeterministickou cestou, potažmo pro ověření jedinečnosti řešení. Vstupními parametry jsou *valuesInIn* – zadání mřížky a *numberOfSolutionsIn* – počet hledaných řešení.

Vzhledem k velké výpočetní náročnosti je nejprve provedeno „předřešení“ deterministickými metodami funkce *candidatesSolutions*, která do mřížky ve většině případů doplní alespoň nějaká čísla, což značně zmenší počet možných řešení, kterými bude *backtracking* procházet a výpočet tak několikanásobně urychlí.

Voláním funkce s hodnotou 1 v parametru *numberOfSolutionsIn* funkce vrátí první úplné řešení, na které narazí. Vstupní hodnota 2 se používá pro ověření jedinečnosti řešení vůči danému zadání. Nalezení dvou různých řešení indikuje jeho nejednoznačnost, pouze jedno znamená jednoznačnost a nula říká, že pro toto zadání žádné řešení neexistuje.

4.2.3 Metoda pro vytváření zadání

Generování zadání zajišťuje funkce *generateAssignment*, jejíž jediný vstupní parametr *difficulty* reprezentuje volbu obtížnosti.

Vzhledem k obrovské výpočetní náročnosti dané nutností ověřování jednak jedinečnosti řešení metodou *backtracking*, a jednak splnění požadavků vůči zvolené obtížnosti, není přijatelné vytvářet nové zadání na „zelené louce“. Rozhodl jsem se tedy vzít jedno kompletní řešení, ze kterého lze pomocí určitých úprav zajišťujících ponechání platnosti a jedinečnosti řešení vytvořit velké množství naprosto odlišných kombinací, mezi kterými uživatel nemá šanci vysledovat spojitost.

Mezi zmíněné pozměňující úpravy patří libovolné promíchání trojic řádků (resp. sloupců) v rámci celé mřížky, tří řádků (resp. sloupců) v rámci každé trojice a transponování celé matice (mřížky). Takto lze z jednoho kompletního řešení velmi rychle vytvořit 23 328 ($3! \cdot 3! \cdot 3! \cdot 3! \cdot 3! \cdot 3! \cdot 2$) různých splňujících všechna pravidla sudoku. Počet je možno dále navýšit úvodním výběrem upravovaného řešení např. z deseti přednastavených.

Samotný algoritmus vytváření zadání z úplného řešení pak spočívá v určení náhodného pořadí, v jakém bude docházet k pokusům o odstraňování čísel z řešení. V cyklu je zapamatována odebíraná hodnota a ověřena jedinečnost řešení a splnění podmínek obtížnosti mřížky bez tohoto čísla. Daná hodnota je v závislosti na výsledku testu buď vrácena zpět, anebo ponechána odstraněna. Pokud ani po pokusu o odstranění poslední hodnoty zadání nespĺňuje požadavky dané obtížností, pak je toto zadání zahozeno a celý postup včetně pozměňujících úprav opakován znovu, dokud výsledné zadání nevyhovuje podmínkám obtížnosti.

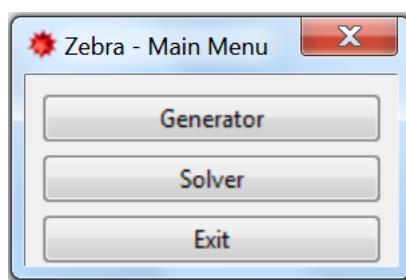
Počet výsledných zadání je tedy mnohem větší než 23 328, protože pro každé řešení lze vytvořit nespočet různých zadání vedoucích ke stejnému cíli. Dvě různá zadání se stejným řešením by uživatel na první pohled jistě nepoznal, nicméně šance pro vygenerování dvou totožných řešení po sobě je jedna ku 23 328, tedy opravdu malá.

5 ZEBRA PUZZLE

Program *Zebra* poskytuje uživateli příjemné interaktivní prostředí pro vytváření různých obtížností zadání dle uživatelsky plně nastavitelných parametrů nebo vyřešení jakékoliv zebra puzzle, jejíž zadání je možno programu sdělit pomocí vestavěných symbolických značek.

5.1 Uživatelský manuál

Obrázek 16 níže uživatel uvidí na své obrazovce ihned po spuštění programu.

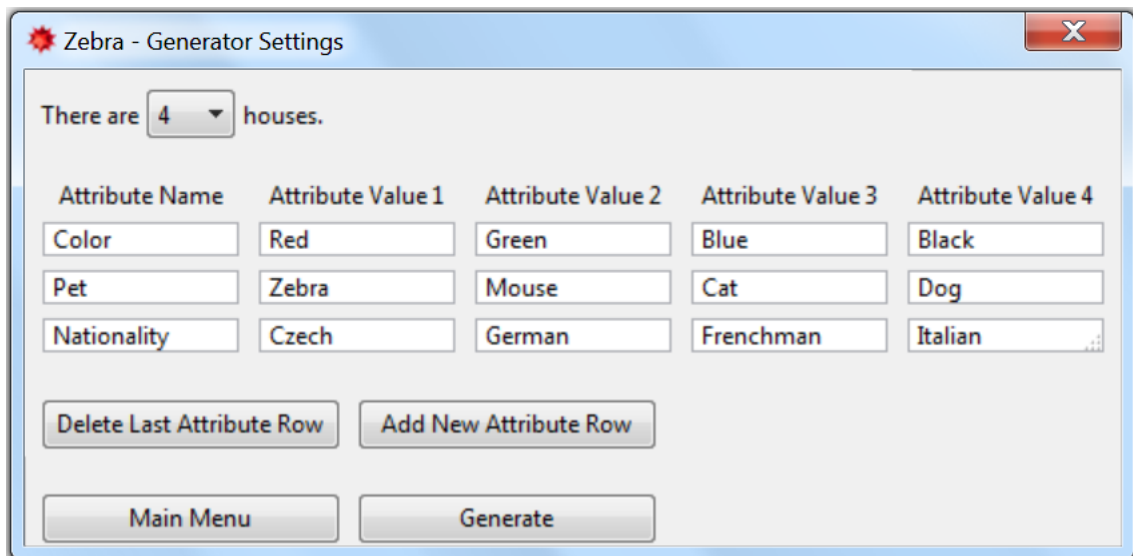


Obrázek 16 Hlavní menu
programu Zebra

Nabídka obsahuje tlačítka *Generator* – otevře okno s nastavením pro vytvoření zadání, *Solver* – vyvolá konfigurační dialog pro vyřešení zadání od uživatele a *Exit* – ukončí program *Zebra*.

5.1.1 Vytváření zadání pro uživatele

Kliknutí na tlačítko *Generator* v hlavním menu otevře interaktivní okno, jehož vzhled závisí na zvolených parametrech. Následující dialog (viz Obr. 17) bude po kliknutí na tlačítko *Generate* vytvářet zadání se čtyřmi domy a třemi neznámými vlastnostmi domů.

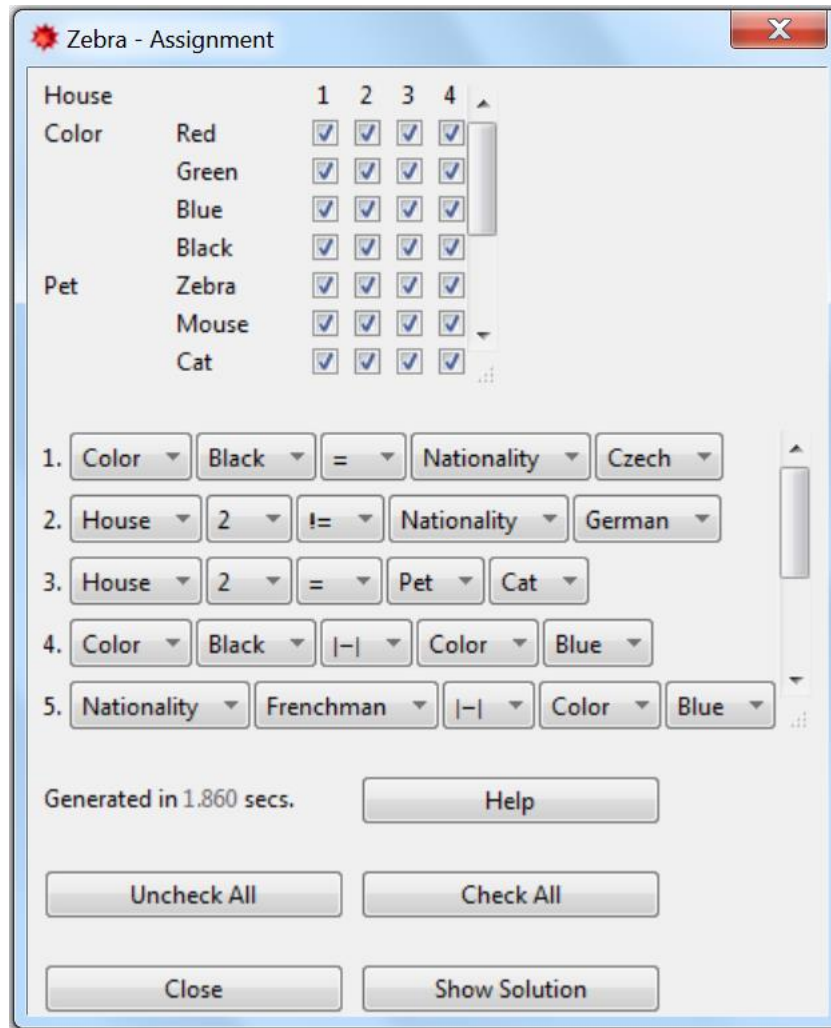


Obrázek 17 Konfigurační dialog pro vytváření zadání

- *Rozevírací seznam* – Představuje možnost volby počtu domů v rozmezí od tří po deset.
- *Upravovatelná textová pole* – Umožňují uživateli vlastní pojmenování názvů a hodnot vlastností domů. Žádné pole nesmí zůstat prázdné a všechna jména musí být v rámci celé tabulky jedinečná.
- *Tlačítko Delete Last Attribute Row* – Smaže poslední řádek tabulky vlastností domů. Tabulka musí vždy obsahovat alespoň jeden řádek.
- *Tlačítko Add New Attribute Row* – Přidá nový řádek do tabulky vlastností domů. Maximální počet řádků tabulky je deset.
- *Tlačítko Main Menu* – Návrat do hlavního menu. Nastavené parametry si program nebude pamatovat.
- *Tlačítko Generate* – Vytvoří zadání s jedinečným řešením dle zvolených parametrů a otevře jej v novém okně. Čím větší počet domů nebo čím větší množství neznámých vlastností, tím vyšší obtížnost a tím delší dobu vytváření potrvá. V závislosti na HW a „štěstí“, na kolikátou iteraci se podaří nalézt zadání splňující všechny požadavky, se celková doba pohybuje od setin sekundy pro minimální parametry výše. Maximální čas, po který se program snaží vytvořit zadání a „neodpovídá“, je stanoven na 60 sekund.

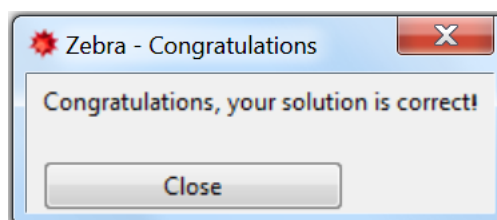
5.1.2 Řešení vytvořených zadání uživatelem

Tlačítko *Generate* v konfiguračním okně pro vytváření zadání otevře interaktivní dialog, který může pro čtyři domy a tři neznámé vlastnosti vypadat například jako Obrázek 18.



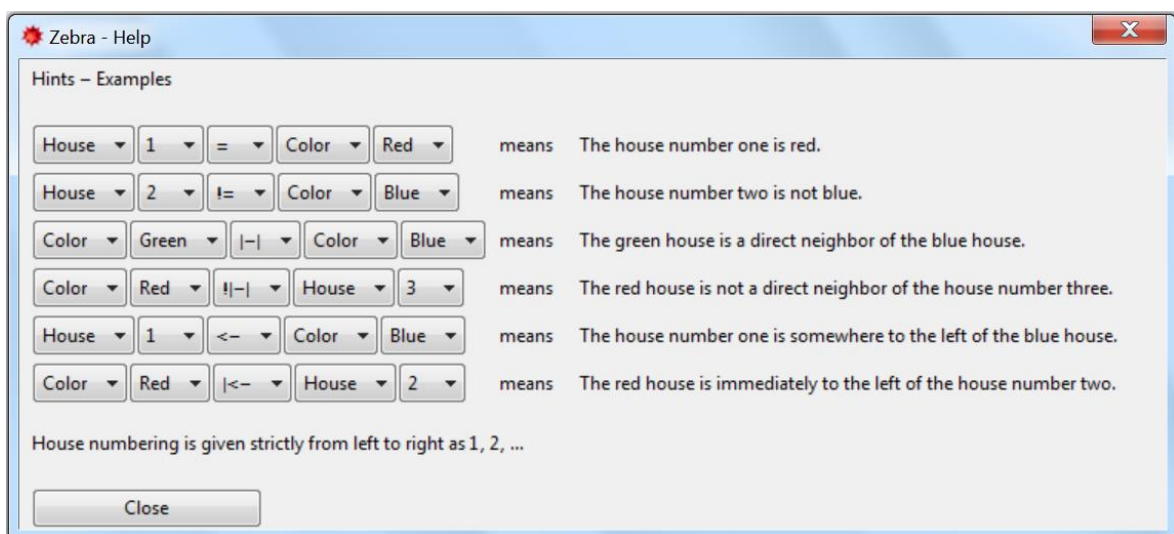
Obrázek 18 Příklad vygenerovaného zadání se čtyřmi domy a třemi neznámými vlastnostmi

- *Tabulka zaškrťovacích políček* – Uživatel ponechá v každém řádku a v každém sloupci „subtabulky“ pro danou vlastnost (obecně rozměru $n \times n$, kde n je počet domů – v případě na Obrázku 18 tedy rozměru 4×4) právě jedno zaškrtnuté políčko dle jeho domněnky o správném řešení. Automatická kontrola případně vyvolá okno s blahopřáním (viz Obr. 19).



Obrázek 19 Okno s blahopřáním

- *Tabulka se zadáním* – Obsahuje vodítka pro vyřešení hádanky. Rozevírací seznamy jsou použity pouze pro větší přehlednost a nelze je rozevřít. Význam vodítek viz nápověda.
- *Informace Generated In* – Udává celkový čas ve vteřinách spotřebovaný pro vytvoření aktuálního zadání.
- *Tlačítko Help* – Vyvolá okno s nápovědou na Obrázku 20. Rozevírací seznamy jsou fiktivní a slouží pouze pro lepší přehlednost.



Obrázek 20 Nápověda programu Zebra

- *Tlačítko Uncheck All* – Odškrtně všechna zaškrťovací políčka.
- *Tlačítko Check All* – Zaškrtně všechna zaškrťovací políčka.
- *Tlačítko Close* – Návrat do konfiguračního dialogu pro vytváření zadání, který si pamatuje aktuálně nastavené parametry.
- *Tlačítko Show Solution* – V tabulce zaškrťovacích políček zobrazí správné řešení.

5.1.3 Řešení zadání od uživatele

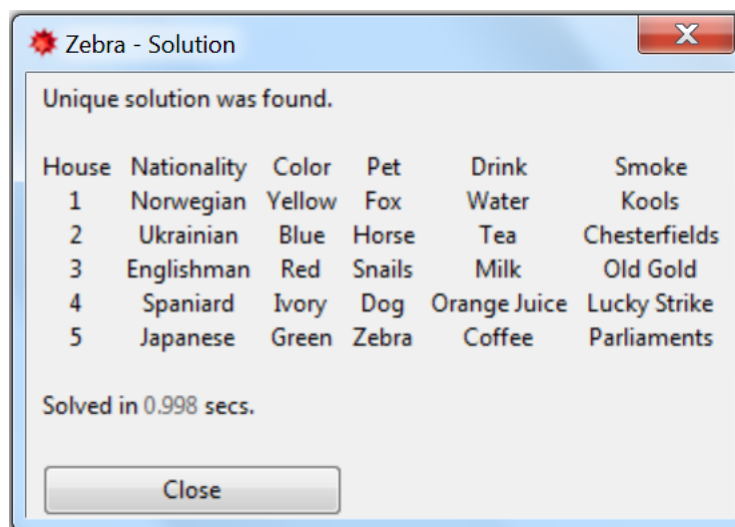
Kliknutí na tlačítko *Solver* v hlavním menu otevře interaktivní okno, jehož obsah je odvislý od zvolených parametrů. Obrázek 21 níže reprezentuje původní Einsteinovu hádanku, jejíž text se nachází v teoretické části.

Obrázek 21 Zadání klasické Einsteinovy hádanky

Většina ovládacích prvků konfiguračního dialogu pro řešení zadání od uživatele již byla popsána v předcházejících dvou podkapitolách, proto jsou zde popsány pouze ty odlišné.

- *Rozevírací seznam* – Viz kapitola 5.1.1.
- *Upravovatelná textová pole* – Viz kapitola 5.1.1.
- *Tlačítko Delete Last Attribute Row* – Viz kapitola 5.1.1. Navíc nesmí být žádná položka z posledního řádku použita v tabulce vodítek, jinak tento řádek není možné smazat.
- *Tlačítko Add New Attribute Row* – Viz kapitola 5.1.1.
- *Rozevírací seznamy* – Reprezentují počítači srozumitelnou formu textových vodítek. V prvním a čtvrtém sloupci se vybírá název vlastnosti, ve druhém a pátém pak hodnota vlastnosti vzhledem k vybranému názvu a ve třetím sloupci vztah mezi těmito vlastnostmi. Význam vztahů viz nápověda.
- *Tlačítko Delete Last Hint Row* – Smaže poslední řádek tabulky vodítek. Tabulka musí vždy obsahovat alespoň jeden řádek.

- *Tlačítko Add New Hint Row* – Přidá nový prázdný řádek do tabulky vodítek. Maximální počet řádků tabulky není omezen.
- *Tlačítko Main Menu* – Viz kapitola 5.1.1.
- *Tlačítko Help* – Viz kapitola 5.1.2.
- *Tlačítko Solve* – Vyvolá okno s výsledkem řešitele. Pokud bylo nalezeno alespoň jedno řešení splňující všechna omezení, pak je toto vypsáno do přehledné tabulky a pokud bylo toto řešení jediným, je prohlášeno za unikátní (viz Obr. 22). V případě více možných řešení je vypsán jejich počet a zobrazeno první z nich. Pokud žádné řešení neexistuje, je o tom uživateli podána zpráva. Čím větší počet domů nebo čím větší množství neznámých vlastností, tím déle bude výpočet trvat. Na druhou stranu větší počet vodítek (zejména typu přiřazení („=“) nebo vyloučení („!=“)) dobu značně zkrátí. Pro daný HW a zvolené parametry je rychlost výpočtu stabilní a nezáleží na „štěstí“. Maximální čas, po který se program snaží dobrat výsledku a „neodpovídá“, je stanoven na 60 sekund.



Obrázek 22 Řešení klasické Einsteinovy hádanky

5.2 Programátorský manuál

Následující podkapitoly vysvětlují techniky řešení a vytváření zadání programu *Zebra*.

5.2.1 Metoda řešení

Vstupními parametry funkce *solver* jsou *modeIn* – testuje-li se pouze jedinečnost řešení při generování nového zadání, anebo má být výsledek i vypsán, *housesCountIn* – počet domů, *attributesMatrixIn* – tabulka názvů a hodnot vlastností a *hintsMatrixIn* – tabulka vodítek.

Vytvoří se h^a a -tic, kde h je počet domů a a počet vlastností. Pro klasickou Einsteinovu hádanku s pěti domy a šesti vlastnostmi (pět neznámých a jedna známá (čísla domů)) je tedy třeba vytvořit 5^6 , resp. 15 625 šestic, z nichž každá představuje jednu možnou kombinaci vlastností jednoho z domů. Počet všech potenciálních řešení lze tedy vyjádřit kombinačním číslem $\binom{h^a}{h}$, v tomto případě $\binom{15625}{5}$, tedy přibližně $7,8 \cdot 10^{18}$ možných řešení, přičemž ověřování každého z nich na splnění všech podmínek není v reálném čase proveditelné. Je proto zapotřebí vydat se nejprve cestou testování všech a -tic a vyřazení z nich všech nepřípustných kombinací vzhledem k vodítkům s relacemi přiřazení („=“) a vyloučení („!=“). Pro tento úkon se skvěle hodí tzv. *Mathematica patterns* a vestavěná funkce *DeleteCases*.

Ze zbylých a -tic se vytvoří kombinace všech možných h a -tic (kompletních sad vlastností všech domů) tak, aby se zde každá vlastnost vyskytovala právě jednou.

Nakonec jsou z kompletních sad vyřazeny ty, které nesplňují zbývající podmínky týkající se sousedství a směrů.

5.2.2 Metoda vytváření zadání

Vstupními parametry funkce *generator* jsou *modeIn* – s touto hodnotou se volá funkce *solver* tak, aby vrátila pouze počet nalezených řešení a nevykreslovala výsledek do nového okna, *housesCountIn* – počet domů, *attributesMatrixIn* – tabulka názvů a hodnot vlastností a *hintsRelationsIn* – symbolické značky relací.

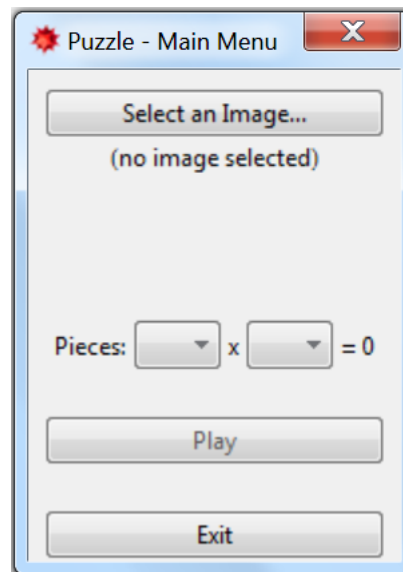
Nejprve jsou náhodně zamíchány hodnoty vlastností domů a vytvořena správná kombinace označených zaškrtačkových políček. Následně je vytvořeno $h \cdot a$ pravdivých vodítek, kde h je počet domů a a počet vlastností. Pomocí funkce *solver* se zjistí, jestli vygenerovaná sada vodítek postačuje pro vymezení jedinečného řešení – pokud ne, vodítka jsou zahozena a vygenerována znovu, což se provádí, dokud není dosaženo sady vedoucí na jedinečné řešení. Nakonec je testována unikátnost řešení postupně bez každého z vodítek a všechna redundantní vodítka mazána, aby výsledné zadání bylo minimální.

6 PUZZLE

Aplikace nabízí uživateli jednoduché nastavení parametrů puzzle a intuitivní ovládání při jeho skládání.

6.1 Uživatelský manuál

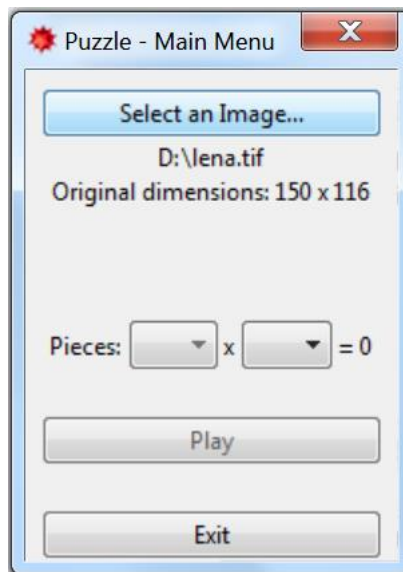
Po spuštění programu uživatel uvidí následující Obrázek 23.



Obrázek 23 Výchozí stav
hlavního menu programu
Puzzle

Nejprve je nutné načíst obrázek, který chce uživatel skládat. Tlačítko *Select an Image* otevře nativní dialogové okno příslušného operačního systému pro načtení souboru. Výchozím adresářem je složka, ze které byla aplikace spuštěna. Podporované přípony jsou *bmp*, *gif*, *jpg*, *jpeg*, *pcx*, *png*, *tga*, *tif* a *tiff*.

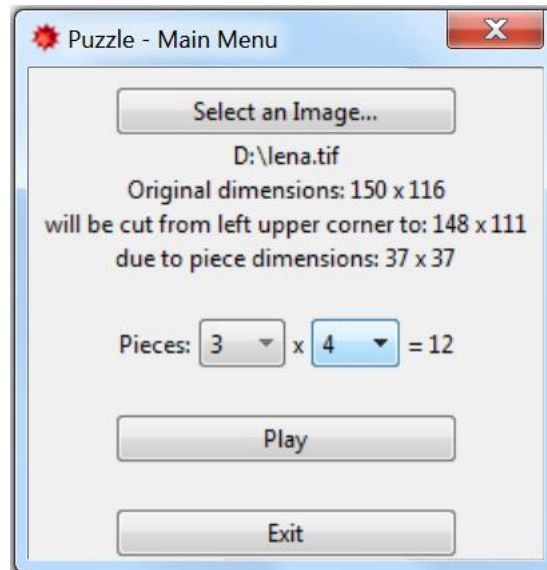
Zpětnou vazbou po načtení platného obrázku je zobrazení informací o jeho umístění na disku a jeho původních rozměrech ve tvaru $w \times h$, kde w je šířka a h výška (viz Obr. 24).



Obrázek 24 Hlavní menu po úspěšném načtení obrázku

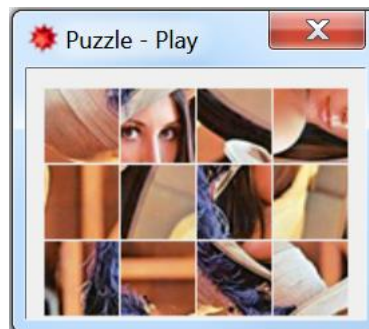
Dále je zapotřebí zvolit počet dílků, na které má být obrázek rozřezán. Množství dílků je zobrazeno ve tvaru $r \times c = s$, kde r je počet řádků, c počet sloupců a s celkový počet dílků. Vzhledem k tomu, že bude obrázek rozdělen na čtvercové části, je možno volit pouze počet dílků delší strany obrázku. Množství úseků kratší strany se automaticky dopočítává. Výjimkou jsou čtvercové obrázky, které budou vždy rozděleny na stejný počet řádků a sloupců, a proto je možné volit libovolnou stranu.

Vybráním počtu dílků získá uživatel informace o celkovém rozměru skládaného obrázku (viz Obr. 25), který bude vzhledem k rozměrům jednoho čtvercového dílku ve většině případů menší oproti původnímu. K ořezání originálního obrázku dochází zespoda a zprava.



Obrázek 25 Hlavní menu po zvolení počtu dílků

Tlačítko *Play* vyvolá okno s promíchanými a různě natočenými dílky rozřezaného obrázku (viz Obr. 26).



Obrázek 26 Zadání skládky

Cílem hry je složení nařezaného obrázku do původní podoby. Dílky lze otáčet nebo přemisťovat. Nejdříve uživatel klikne myší na část, se kterou chce pracovat. Označený čtvereček získá modrý lem. Druhé kliknutí na stejný dílek jej otočí o 90 stupňů po směru hodinových ručiček. Pokud však podruhé klikne na jiný čtvereček, dojde k jejich výměně.

Automatická kontrola okamžitě po zjištění správného výsledku otevře okno s blahopřáním a složeným obrázkem (viz Obr. 27).



Obrázek 27 Okno s
blahopřáním

6.2 Programátorský manuál

Zvolením množství dílků delší strany c jsou určeny rozměry jednoho kusu obrázku. Strana čtverečku p se spočítá jako celá část výrazu $\frac{l}{c}$, kde l je velikost delší strany obrázku a c počet jejích dílků. Množství částí kratší strany je pak dáno jako celá část $\frac{s}{p}$, kde s je rozměr kratší strany obrázku a p délka strany čtverečku. V nejhorším případě tedy dojde k ořezání obrázku o $c-1$ pixelů na jeho delší straně, kde c je počet dílků delší strany, a o $p-1$ pixelů na straně kratší, kde p je délka strany čtverečku. U čtvercových obrázků hrozí na obou stranách maximální ztráta $c-1$ pixelů a obecně lze říci, že čím více se budou rozměry obrázku blížit čtverci, tím méně bude z obrázku vynecháno. V nejlepší situaci obrázek vůbec ořezáván nebude.

Alternativním postupem bez zkracování by bylo „vycpávání“ (např. bílým pozadím), kde okrajové dílky přesahují původní obrázek. Určení pozice čtverečků s bílým lemem by však bylo pro uživatele velmi snadné, a proto byla tato cesta zavržena.

ZÁVĚR

Spustitelné zdrojové kódy pro SW Mathematica ke zmíněným logickým hrám se nachází na příloženém CD.

Pro mezinárodní srozumitelnost byla jazykem uživatelských prostředí všech her, ale i poznámek a názvů proměnných a funkcí ve zdrojových kódech, zvolena angličtina. Zdrojové kódy jsou přehledně členěny a jejich důležité části stručně okomentovány. Jména funkcí a proměnných byla volena tak, aby i za cenu delšího názvu co možná nejvíce vystihovala jejich účel.

Hra *Sudoku* klade důraz na interaktivitu, přehlednost a uživatelské pohodlí podpořené množstvím funkčních prvků umožňujících vytváření čtyř druhů obtížnosti zadání, řešení úloh od uživatele různými kombinacemi tří řešících technik s možností krokování pro lepší pochopení jejich principu, ověření jedinečnosti řešení daného zadání bez nutnosti jeho vyzrazení, možnost nápovědy v podobě zobrazení zbývajících kandidátů a v neposlední řadě možnost uložení rozřešené mřížky do souboru na pevný disk s možností jeho opětovného načtení.

Návrhem na vylepšení ve formě zkrácení čekacích časů na vygenerování zadání vyšších obtížností či otestování unikátnosti by mohlo být implementování pokročilejších deterministických řešících metod. Úzké hrdlo totiž představuje technika *backtracking*, jejíž časová náročnost závisí na počtu čísel předvyplněných ze zadání či „předpočítaných“ právě deterministickými metodami. Dále by bylo možné výměnou několika konstant za proměnné zobecnit rozměr mřížky.

Aplikace *Zebra* poskytuje dynamické prostředí pro generování různých zadání zebra puzzle, kde stupeň obtížnosti závisí na uživatelem nastavených parametrech. Dále obsahuje řešitele uživatelem vloženého zadání, kde jsou relace mezi vlastnostmi interpretovány sadou symbolických značek. Volitelnými parametry jsou počet domů a také množství i jména jejich vlastností.

Námětem pro vylepšení je vymyšlení rychlejšího způsobu pro nalezení řešení, což by se promítlo ve vytváření zadání s větším množstvím domů či vlastností během kratší doby.

Program *Puzzle* nabízí možnost skládání libovolného obrázku v podporovaném formátu načteného z pevného disku. Obtížnost je dána jednak počtem čtvercových dílků, na které si

uživatel nechá obrázek nařezat, a jednak tím, jestli uživatel obrázek předem viděl, či nikoliv. Ovládání hry při skládání a otáčení dílků je snadné a intuitivní.

Velmi ambiciózním vylepšením by bylo rozšíření puzzle o třetí rozměr. Vzhledem k tomu, že Mathematica podporuje práci s 3D obrázky, byl by tento nápad snad i realizovatelný.

Domnívám se, že se mi podařilo vytvořit ucelenou knihovnu rozmanitých logických her, které uživatelům poslouží jak během přípravy na testy logického myšlení, tak i jako volnočasová kreativní a inteligentní zábava.

SEZNAM POUŽITÉ LITERATURY

- [1] HLADÍK, Jaroslav. *Společenské vědy v kostce: pro střední školy*. 3. vyd. Praha: Fragment, 2004, 108 s. ISBN 978-807-2009-664.
- [2] JIRKŮ, Petr a Jiřina VEJNAROVÁ. *Formální logika: neformální výklad základů formální logiky*. Vyd. 1. Praha: Vysoká škola ekonomická, Fakulta informatiky a statistiky, 2000, 161 s. ISBN 978-802-4500-546.
- [3] VLČEK, Miroslav. *Úvod do klasické (aristotelovské) logiky*. Vyd. 1. Praha: Vysoká škola ekonomická, 1996, 114 s. ISBN 80-707-9799-1.
- [4] BOREK, Bernard. Odkaz Aristotelové logiky. In: *Píše a tvoří Borek Bernard* [online]. 2004 [cit. 2014-06-01]. Dostupné z: <http://www.borber.com/materialy/skola/fil103-axiologie>
- [5] JIRKŮ, Petr a Jozef KELEMEN. *Kapitoly z kognitivní vědy: racionalita z hlediska chování, jazyka a logiky*. Vyd. 1. Praha: Vysoká škola ekonomická v Praze, 1996, 104 s. ISBN 80-707-9787-8.
- [6] HARTL, Pavel a Helena HARTLOVÁ. *Psychologický slovník*. Vyd. 2. Praha: Portál, 2009, 774 s. ISBN 978-807-3675-691.
- [7] BERTHIER, Denis. *The hidden logic of Sudoku*. S.l.: Lulu.com, 2007, 384 s. ISBN 978-184-7534-729.
- [8] BOYER, Christian. Sudoku's French ancestors. *Multimagie* [online]. 2007 [cit. 2014-04-30]. Dostupné z: <http://web.archive.org/web/20071010081626/http://cboyer.club.fr/multimagie/English/SudokuAncestors.htm>
- [9] PEGG, Ed, Jr. Math Games: Sudoku Variations. *Mathpuzzle* [online]. 2005 [cit. 2014-05-02]. Dostupné z: http://www.mathpuzzle.com/MAA/41-Sudoku%20Variations/mathgames_09_05_05.html
- [10] YATO, Takayuki a Takahiro SETA. Complexity and Completeness of Finding Another Solution and Its Application to Puzzles. In: *Takayuki Yato's Site* [online]. 2002 [cit. 2014-05-05]. Dostupné z: <http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/SIGAL87-2.pdf>
- [11] JARVIS, Frazer. Sudoku enumeration problems. *Frazer Jarvis's home page* [online]. 2008 [cit. 2014-05-06]. Dostupné z: <http://www.afjarvis.staff.shef.ac.uk/sudoku/>

- [12] MCGUIRE, Gary, Bastian TUGEMANN a Gilles CIVARIO. There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem. *Cornell University Library* [online]. 2012 [cit. 2014-05-06]. Dostupné z: <http://arxiv.org/abs/1201.0749v2>
- [13] A nonomino sudoku. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2008 [cit. 2014-06-05]. Dostupné z: https://en.wikipedia.org/wiki/Sudoku#mediaviewer/File:A_nonomino_sudoku.svg
- [14] STANGROOM, Jeremy. *Einstein's riddle: riddles, paradoxes, and conundrums to stretch your mind*. 1st U.S. ed. New York: Bloomsbury USA, 2009, 144 p. ISBN 978-1-59691-665-4.
- [15] Who Owns the Zebra?: A Problem for the Logical. *Life International*. 1962. Dostupné z: <https://upload.wikimedia.org/wikipedia/en/8/82/Zebra.abridged.jpg>
- [16] Zebra Puzzle. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-05-11]. Dostupné z: https://en.wikipedia.org/wiki/Zebra_puzzle
- [17] Why Jigsaw Puzzles Are Good For The Brain. *Social Psychiatry Blog* [online]. [cit. 2014-05-13]. Dostupné z: <http://social-psychiatry.com/jigsaw-puzzles-good-brain/>
- [18] MCADAM, Daniel. History of Jigsaw Puzzles. *American Jigsaw Puzzle Society: An online guide to jigsaw puzzles and resources* [online]. [cit. 2014-05-14]. Dostupné z: <http://www.jigsaw-puzzle.org/jigsaw-puzzle-history.html>
- [19] Wikipedia-logo-v2. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2011 [cit. 2014-06-05]. Dostupné z: http://en.wikipedia.org/wiki/Logo_of_Wikipedia#mediaviewer/File:Wikipedia-logo-v2.svg
- [20] Ignis-spindle-fin12. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012 [cit. 2014-06-05]. Dostupné z: https://en.wikipedia.org/wiki/Jigsaw_puzzle#mediaviewer/File:Ignis-spindle-fin12.jpg
- [21] JigsawSoftware. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2011 [cit. 2014-06-05]. Dostupné z:

https://en.wikipedia.org/wiki/Jigsaw_puzzle#mediaviewer/File:JigsawSoftware.jpeg

- [22] Largest commercial jigsaw puzzle: most pieces. In: *Guinness World Records* [online]. 2010 [cit. 2014-05-15]. Dostupné z: <http://www.guinnessworldrecords.com/world-records/2000/largest-commercial-jigsaw-puzzle-most-pieces>
- [23] Largest jigsaw puzzle. In: *Guinness World Records* [online]. 2002 [cit. 2014-05-15]. Dostupné z: <http://www.guinnessworldrecords.com/world-records/1/largest-jigsaw-puzzle->
- [24] What can webmathematica do that CDF cannot do?. In: *Mathematica: Stack Exchange* [online]. 2013 [cit. 2014-05-18]. Dostupné z: <http://mathematica.stackexchange.com/questions/21601/what-can-webmathematica-do-that-cdf-cannot-do/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- 3D Trojdimenzionální, trojrozměrný.
- BT Backtracking – metoda zpětného vyhledávání, metoda pokusů a oprav, metoda prohledávání do hloubky.
- CDF Computable Document Format – typ souboru spustitelného v kterémkoliv Wolfram CDF přehrávači.
- CSV Comma Separated Values – typ souboru tabulkových dat, ve kterém jsou položky odděleny čárkou.
- HS Hidden Singles – technika řešení sudoku.
- HTML HyperText Markup Language – hypertextový značkovací jazyk pro tvorbu webových stránek.
- HW Hardware – fyzicky existující technické vybavení počítače.
- LSAT Law School Admission Test – mezinárodní zkouška logických schopností, na které je založeno přijímací řízení mnoha právnických univerzit po celém světě.
- MS Microsoft – americká společnost zabývající se počítačových průmyslem.
- MSP Mathematica Server Pages – základní technologie SW webMathematica obdobná technologii Java Server Pages.
- NS Naked Singles (zkráceně také Singles) – technika řešení sudoku.
- S Singles (také Naked Singles) – technika řešení sudoku.
- SCIO Scio – česká společnost zabývající se testováním schopností žáků základních a středních škol.
- SW Software – programové vybavení počítače.

SEZNAM OBRÁZKŮ

Obrázek 1 Příklad zadání sudoku	16
Obrázek 2 Řešení zadání sudoku vlevo	16
Obrázek 3 Úvodní výpis všech kandidátů	18
Obrázek 4 Kandidáti po aplikaci elementárních pravidel.....	18
Obrázek 5 Příklad „Naked Single“ v prostředním bloku.....	20
Obrázek 6 Doplnění „Naked Single“ z obrázku vlevo	20
Obrázek 7 Příklad „Hidden Single“ v levém prostředním bloku.....	20
Obrázek 8 Doplnění „Hidden Single“ z obrázku vlevo	20
Obrázek 9 Příklad zadání sudoku s nepravidelnými bloky[13].....	21
Obrázek 10 Logo Wikipedie[19]	27
Obrázek 11 Vrstvené trojrozměrné puzzle[20].....	27
Obrázek 12 Software ke stažení zdarma pro skládání puzzle umožňující rotaci dílků[21]	27
Obrázek 13 Výchozí stav dialogového okna po spuštění programu Sudoku	32
Obrázek 14 Ověření jedinečnosti řešení pro zadání v mřížce	34
Obrázek 15 Náповěda zbývajících kandidátů u nevyplněných polí.....	35
Obrázek 16 Hlavní menu programu Zebra	38
Obrázek 17 Konfigurační dialog pro vytváření zadání.....	39
Obrázek 18 Příklad vygenerovaného zadání se čtyřmi domy a třemi neznámými vlastnostmi.....	40
Obrázek 19 Okno s blahopřáním	40
Obrázek 20 Náповěda programu Zebra.....	41
Obrázek 21 Zadání klasické Einsteinovy hádanky	42
Obrázek 22 Řešení klasické Einsteinovy hádanky	43
Obrázek 23 Výchozí stav hlavního menu programu Puzzle.....	45
Obrázek 24 Hlavní menu po úspěšném načtení obrázku	46
Obrázek 25 Hlavní menu po zvolení počtu dílků	47
Obrázek 26 Zadání skládanky.....	47
Obrázek 27 Okno s blahopřáním	48

SEZNAM TABULEK

Tabulka 1 Pravdivostní tabulka implikace.....	12
Tabulka 2 Pravdivostní tabulka disjunkce	12
Tabulka 3 Výchozí stav klasické Einsteinovy hádanky.....	23
Tabulka 4 Výchozí stav zadání se třemi domy a dvěma vlastnostmi	24
Tabulka 5 Předchozí tabulka po aplikaci zmíněného vodítka	25

SEZNAM PŘÍLOH

Příloha P I: CD se spustitelnými zdrojovými kódy pro SW Mathematica