



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**VÝUKOVÝ PROGRAM PRO HRANÍ NETRADIČNÍCH
FOREM HRY SUDOKU**

TUTORIAL FOR PLAYING NON-TRADITIONAL FORMS OF SUDOKU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

IVETA STRNADOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2021

Zadání bakalářské práce



Studentka: **Strnadová Iveta**
Program: Informační technologie
Název: **Výukový program pro hraní netradičních forem hry SUDOKU**
Tutorial for Playing Non-Traditional Forms of SUDOKU
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s variantami hraní her SUDOKU a dále s metodami řešení problémů s omezujícími podmínkami.
2. Navrhněte systém, který by poskytoval uživateli přehledný návod, jak by měl postupovat při hře těchto her. Systém by měl běžnému uživateli demonstrovat, jak uvažovat při řešení těchto úloh.
3. Navržený systém implementujte a otestujte na hráčích různých schopností a zkušeností s těmito úlohami.
4. Zhodnoťte dosažené výsledky a navrhněte možná vylepšení, pokud takové identifikujete.

Literatura:

- Russel, S., Norvig, P.: Artificial Intelligence, A Modern Approach, Pearson, 2009

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zbořil František, doc. Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cílem této práce je prozkoumat logickou hádanku sudoku z pohledu uživatele a vytvořit mu nástroj poskytující pomoc při řešení. Zaměřuje se na klasické sudoku i jeho populární variace. Nástroj ve formě webové stránky je vytvořen v Django. Obsahuje prostředí pro řešení sudoku, vzorová zadání, nápovědu, vysvětlené strategie řešení a možnost najít další krok v řešení logickou cestou.

Abstract

This thesis aims to explore aspects of the sudoku riddle from the user's perspective and create a tool to help him solve it. It deals with both classic sudoku and its popular subtypes. The tool is a website application made in Django. It implements an app for solving sudoku with hints, explained sudoku strategies and a tool for finding the next logical step.

Klíčová slova

Sudoku, alternativní typy sudoku, webová aplikace, výukový program, logické řešení sudoku, návody.

Keywords

Sudoku, alternative sudoku types, web application, tutorial, logical sudoku solving, guides.

Citace

STRNADOVÁ, Iveta. *Výukový program pro hraní netradičních forem hry SUDOKU*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. František Zbořil, Ph.D.

Výukový program pro hraní netradičních forem hry SUDOKU

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením docenta Františka Zbořila. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....

Iveta Strnadová

9. května 2021

Poděkování

Ráda bych poděkovala vedoucímu práce, Doc. Ing. Františku Zbořilovi, Ph.D., za vstřícný přístup, cenné rady a ochotu vést tuto práci. Dále bych ráda poděkovala všem, kteří se účastnili testování aplikace a poskytli mi konstruktivní zpětnou vazbu.

Obsah

1	Úvod	3
2	Sudoku	4
2.1	Latinský čtverec	4
2.2	Základní pojmy	4
2.3	Klasické sudoku	5
2.4	Alternativní typy sudoku	5
2.5	Matematika v sudoku	11
2.6	Generování sudoku	12
3	Strategie řešení sudoku	15
3.1	Odstranění přímých kolizí	16
3.2	Naked single	16
3.3	Hidden single	16
3.4	Naked Pairs, Triples and Quads	16
3.5	Hidden Pairs, Triples and Quads	17
3.6	Intersection Removal	18
3.7	X-Wing	21
3.8	Y-Wing	21
3.9	Swordfish	23
3.10	XY-Chain	23
4	Návrh aplikace	25
4.1	Omezení aplikace	25
4.2	Diagram případů užití	26
4.3	Grafické uživatelské rozhraní	29
4.4	Formát sudoku	32
4.5	Řešení sudoku počítačem	32
4.6	Kompatibilita s dalšími aplikacemi	35
5	Implementace aplikace	37
5.1	Přehled implementace	37
5.2	Generování sudoku	39
5.3	Třídy popisující sudoku	40
5.4	Třída aplikující strategie	41
6	Testování aplikace	45
6.1	Zpětná vazba od uživatelů	46

6.2	Změny aplikace na základě zpětné vazby	47
6.3	Další možná vylepšení	47
7	Závěr	48
	Literatura	49
A	Obsah přiloženého média	51
B	Snímky obrazovky	52

Kapitola 1

Úvod

Pro vyřešení sudoku člověku stačí zadání, tužka a papír. Pokud je dostatečně zdatný, zvládne se postupným doplňováním čísel dostat k plně vyplněné mřížce a kontrolou splněných pravidel ověřit, že dospěl k správnému řešení. Problém však nastane, pokud uživatel v průběhu řešení zjistí, že neví jak pokračovat.

V dnešní době se na internetu nachází nepřeberné množství zadání studoku, jejich generátorů, návodů, nástrojů na jejich řešení přímo na stránce či nástrojů na jejich vyřešení. Většina z nich však obsahuje pouze základní funkce a často není uživatelsky příjemná.

Vyřešit sudoku za pomoci počítače či zobrazit uložené správné řešení umí většina nástrojů. Avšak správné řešení uživatele v schopnosti řešit sudoku nikam neposune. Pokud uživatel sudoku vyřešil, nepotřebuje řešení – když v jeho vyřešeném sudoku platí všechna pravidla, je správné. Když uživatel při řešení neví jak pokračovat dál, správné řešení mu neporadí, jak měl postupovat pro objevení dalšího čísla.

Hlavním cílem této práce je vytvořit pro uživatele nástroj, který mu pomůže rozpoznat strategie pro řešení sudoku a použít je v praxi. Pro tyto účely se vyvíjený systém zaměřuje na dva hlavní celky: interaktivní návody popisující jednotlivé strategie a nástroj na řešení zadaného sudoku pomocí postupné aplikace strategií tak, jak postupuje člověk. Systém uživateli pomůže v situaci, když si nebude vědět rady se svým rozluštěným sudoku a bude chtít napovědět, jak pokračovat dál.

Vedlejším cílem je vytvoření prostředí s pohodlným ovládáním pro luštění sudoku a možností ohodnocení obtížnosti zadaného sudoku podle použitých strategií k vyluštění.

Od vzniku sudoku se vyvinulo mnoho jeho podtypů vyžadujících rozšířená pravidla umístění číslic či pracujících s mřížkou o různých velikostech. Najít nápovědu pro tyto druhy je ještě obtížnější a proto se tato práce bude zabývat také pomocí při řešení mnoha rozšířených podtypů sudoku.

V teoretické části jsou prozkoumány vlastnosti sudoku a jeho podtypů, principy rozšířených strategií různých obtížností a teorie generování sudoku. Praktická část tyto znalosti aplikuje do webového nástroje, který klade důraz na pomoc uživateli při řešení sudoku. Tento webový nástroj je dostupný z <https://hojkas-sudoku-helper.herokuapp.com/>.

Kapitola 2

Sudoku

Sudoku je logická hádanka založená na umístování omezeného počtu číslic do mřížky podle předem daných pravidel. Za snadno zapamatovatelným principem se skrývá složitá výzva spočívající v nalezení jediného správného řešení.

2.1 Latinský čtverec

Latinský čtverec o řádu n je pole o rozměrech $n \times n$. Obsahuje n různých symbolů umístěných tak, aby se v každém řádku a sloupci nacházel každý symbol právě jednou.

Ačkoliv název „Latinský čtverec“ použil poprvé až Leonhard Euler v 18. století, čtverce se symboly rozmístěnými pomocí těchto pravidel se dají nalézt již na amuletech datovaných kolem roku 1000 [1].

Ve své podstatě je sudoku latinský čtverec rozšířený o další pravidla pro umístění symbolů.

2.2 Základní pojmy

- Celá plocha sudoku se nazývá *mřížka*, anglicky *grid*.
- Nejmenší jednotce čtverečku se říká *pole*, popř. *políčko* nebo *buňka*, anglicky *cell*.
- Jako *sektor* (anglicky „sector“) se označuje malý čtverec o velikost 3×3 . U sudoků jiných velikostí může jít o čtverec či obdélník jiných rozměrů nebo o nepravidelný útvar. Vždy se jedná o spojení tolika buněk, kolik sudoku obsahuje číslic/symbolů.
- Jeden sloupec, řádek nebo sektor se označuje jako *blok*, anglicky *block*.¹
- Předvyplněná čísla budou označována jako *zadání* nebo *stopy*, anglicky *clues*.
- Možná čísla pro danou buňku nesou název *kandidáty*, popř. *kandidátní čísla*, anglicky *candidates*.

Pro konzistenci s názvoslovím pojmů v češtině byla většina názvů převzata z Do nitra Sudoku [18].

¹Sjednocení pojmenování pro řádek, sloupec a blok má význam pro popis strategií, u kterých většinou nezáleží na tom, jaký tvar má prohledávaný blok.

Pro potřeby popisu konkrétní buňky sudoku zavádím notaci „rXcY“, kde X označuje číslo řádku (počítáno od 1) a Y označuje číslo sloupce (počítáno od 1). Např. r1c1 označuje první řádek první sloupec, tedy levou horní buňku.

Stejná notace bude použita ve výsledné aplikaci pro popis aplikované strategie uživateli.

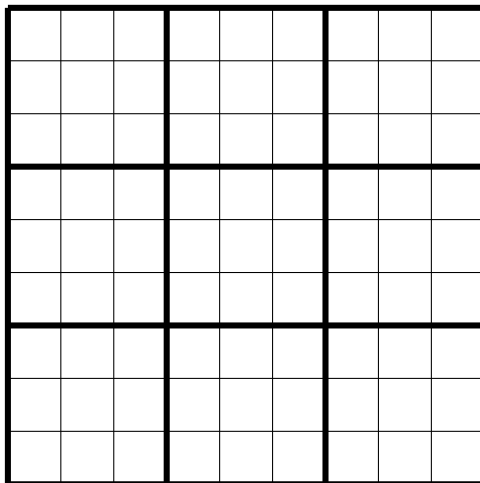
2.3 Klasické sudoku

Mřížka sudoku je o rozměrech 9x9 a je rozdělena na 9 sektorů o velikosti 3x3. Pro čísla definuje tři pravidla, která platí i ve většině dalších typů:

- v každém sloupci musí být každá z číslic 1–9 právě jednou.
- v každém řádku musí být každá z číslic 1–9 právě jednou.
- v každém sektoru musí být každá z číslic 1–9 právě jednou.

Použitím dříve definovaných pojmů lze sloupec, řádek i sektor označit jako blok sudoku. Pravidla tak lze zjednodušit na následující:

- v každém bloku sudoku se musí každá z číslic 1–9 vyskytovat právě jednou.



Obrázek 2.1: Prázdné sudoku. Tlustší čára od sebe odděluje jednotlivé sektory.

2.4 Alternativní typy sudoku

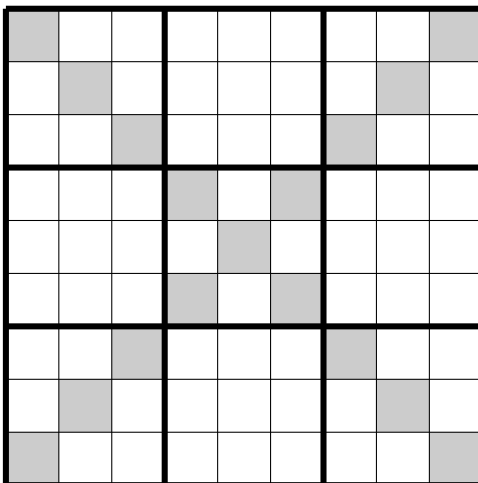
Se sudoku se můžeme setkat v mnoha dalších podobách kromě klasické verze popsané v sekci 2.3. Existují typy sudoku s modifikovanou velikostí hracího pole, tvaru sektorů či přidáním nových pravidel pro umístění číslic.

Možností modifikace sudoku je mnoho a neustále vznikají nové exotické varianty. Pro varianty sudoku neexistuje žádný celosvětově uznávaný standard, který by definoval jejich přesnou podobu a jméno. Názvy a vlastnosti použité v této práci vycházejí z běžně používaných ustálených pojmů.

Diagonální sudoku

Sudoku 9x9 s 9 pravidelnými sektory o rozměrech 3x3. Platí v něm 3 základní pravidla a nové pravidlo o diagonálách:

- v každé diagonále musí být každá z číslic 1–9 právě jednou.



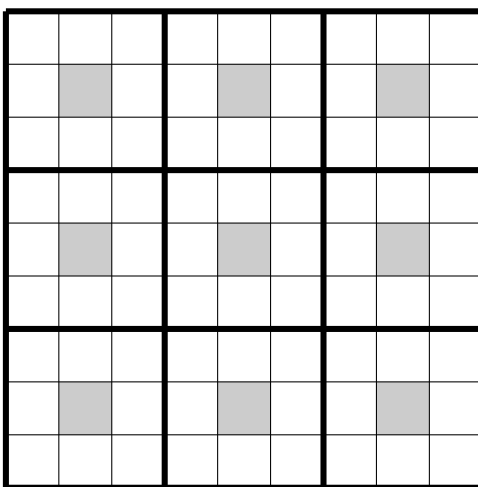
Obrázek 2.2: Prázdné diagonální sudoku. Šedá barva označuje buňky na diagonále.

Sudoku se středy čtverců

Sudoku 9x9 s 9 pravidelnými sektory tvaru čtverců o rozměrech 3x3. Platí v něm 3 základní pravidla a nové pravidlo o středech čtverců:

- v středech čtverců (čtvercových sektorů) se musí vyskytovat každá z číslic 1–9 právě jednou.

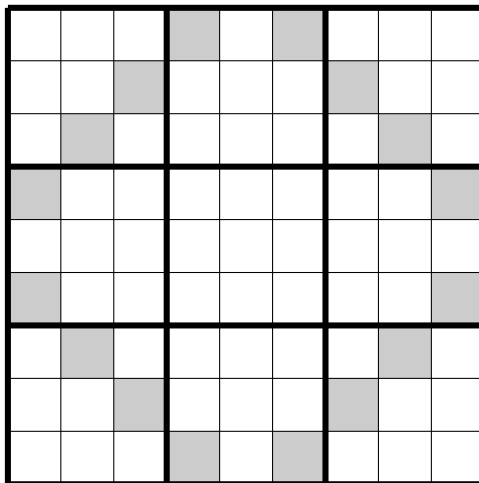
Verze sudoku s pravidlem o středech čtverců je běžně kombinovaná s pravidlem o diagonálách.



Obrázek 2.3: Prázdné sudoku středy čtverců. Šedá barva označuje políčka středů čtverců.

Sudoku s lichými či sudými políčky

Sudoku o velikosti 3×3 , ve kterém platí základní pravidla sudoku. Některé buňky jsou graficky zvýrazněny. Jedná-li se o sudoku s lichými políčky (častější varianta, protože omezí možné kandidáty na 5 možností, sudé omezí pouze na 4), mohou se v těchto buňkách nacházet pouze lichá čísla (1, 3, 5, 7 a 9)². U sudoku se sudými políčky se v těchto buňkách nachází pouze sudá čísla.



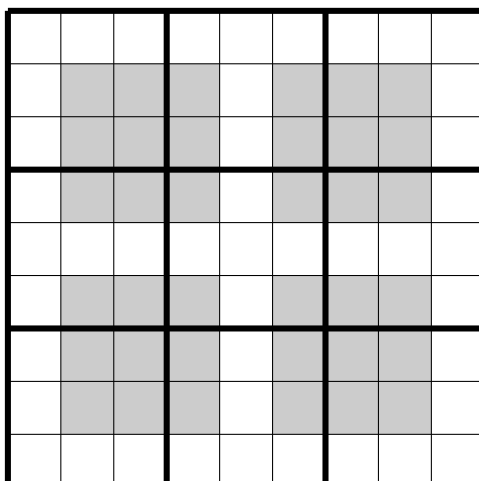
Obrázek 2.4: Prázdné sudoku s lichými políčky. Šedá barva označuje políčka, ve kterých se mohou vyskytovat pouze liché čísla.

Hypersudoku

Sudoku 9×9 s 13 pravidelnými sektory tvaru čtverců o rozměrech 3×3 . Kromě 9 sektorů tvořících základní tvar sudoku jsou v hypersudoku umístěny další 4 čtvercové sektory překrývající se s těmi základními. Pro grafické oddělení bývají nové sektory obvykle zvýrazněny barvou místo tlustého obrysu.

Platí zde 3 základní pravidla.

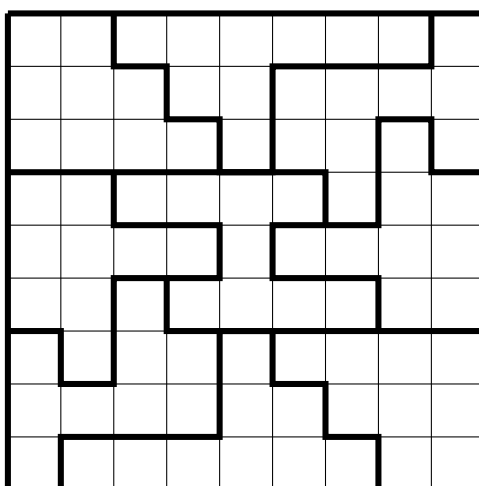
²Tento typ sudoku by se také dal zařadit do sekce 2.4. Avšak omezení na liché/sudé čísla se dá místo matematické podmínky dá chápat pouze jako vytvoření množiny symbolů, které se mohou v buňce nacházet. Sudoku s využitím matematické povahy čísel vyžadují provádění matematických operací a nutí uživatele uvažovat nad sudoku v nových souvislostech. Omezení lichých/sudých čísel oproti nim „pouze“ omezuje počet výchozích kandidátních čísel v buňce. Proto ho uvádím zde.



Obrázek 2.5: Prázdné hypersudoku. Šedá barva značí další sektory překrývající se s těmi základními.

Jigsaw

Sudoku o velikost 9x9. Následuje 3 základní pravidla sudoku. Jeho sektory jsou nepravidelných tvarů.



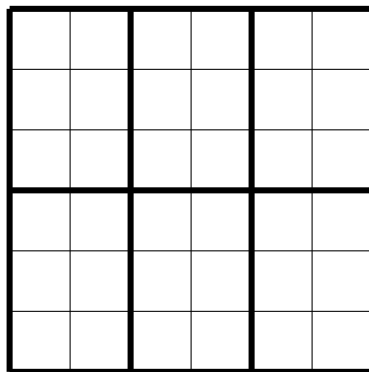
Obrázek 2.6: Prázdné jigsaw sudoku s jedním z mnoha možných rozmístění sektorů.

Netradiční velikosti

Sudoku nemusí být pouze o rozměrech 9x9.

Nejmenší možné sudoku rozdělené na sektory je 4x4 s čtvercovými sektory o velikosti 2x2. Do tohoto sudoku se doplňují čísla 1–4.

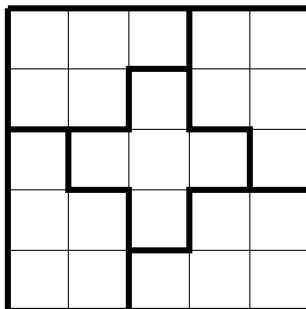
Mezi další rozšířené rozměry sudoku patří 6x6 obsahující obdélníkové sektory 2x3 nebo sudoku 16x16, kam se doplňují čísla 1–16 nebo 1–9 a A–G a sektory jsou čtverce velikosti 4x4.



Obrázek 2.7: Prázdné sudoku o rozměrech 6x6.

Větší velikosti sudoku také existují, ale představují spíše raritu, neboť nejsou kvůli rozsahu komfortní na luštění a logický aspekt hry ustupuje orientaci v obří hrací ploše. Na fórech pro nadšence lze narazit na velikosti 25x25, 100x100 buněk³ či dokonce 225x225.

Další zajímavé tvary sudoku vznikají na poli se sektory nepravidelných tvarů. NxN herní plocha je vyplněna N sektory o velikosti N políček a doplňuje se do ní čísla 1 až N. Tímto stylem vznikají například 5x5 sudoku nebo 7x7 sudoku. (Jigsaw sudoku 9x9 je také příkladem takto vytvořeného sudoku.)

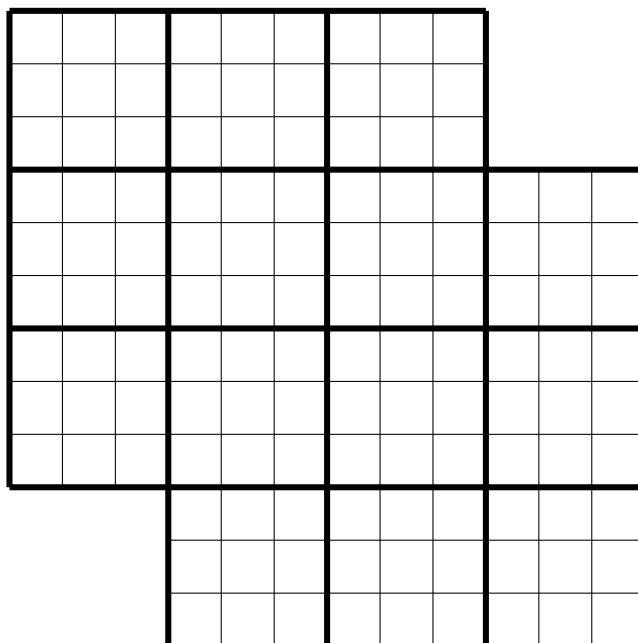


Obrázek 2.8: Prázdné sudoku o rozměrech 5x5.

Velká sudoku lze také vytvořit skládáním menších, obvykle klasických 9x9 sudoku. Největší vyluštěné sudoku se skládalo z 280 spojených sudoku.⁴

³Příklad zadání sudoku 100x100 na fóru: <http://forum.enjoysudoku.com/a-new-100x100-t6395.html>. Sudoku 100x100 bylo také roku 2010 vydáno v tištěné podobě pod jménem *Sudoku-zilla*.

⁴Podle rekordu o největším vyřešeném multi-sudoku z Guinessovy knihy rekordů [7].



Obrázek 2.9: Prázdné sudoku složené ze dvou klasických.

Typy sudoku využívající matematické povahy čísel

Doteď zmíněné typy sudoku využívaly čísla pouze jako návzájem různé symboly. Existují druhy sudoku, které využívají matematické vlastnosti doplňovaných čísel.

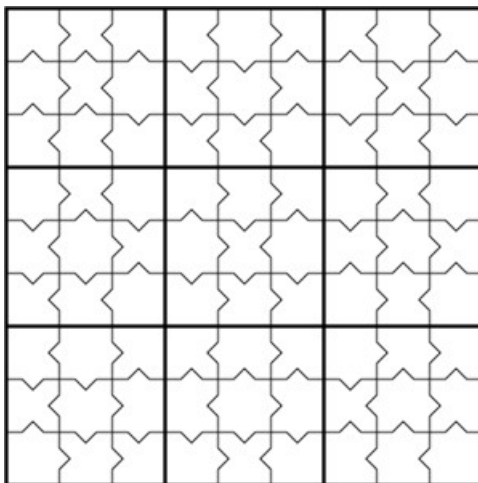
Tyto obměny hádanky představují dodatečnou výzvu a nutí řešitele nad čísly uvažovat v novém rozměru. Na druhou stranu se kvůli tomu ztrácí jednoduchost pravidel sudoku a provádět během luštění matematické operace může být odrazující.

Killer sudoku bývá v češtině označováno také jako „sčítací“. Platí v něm 3 základní pravidla jako v klasickém sudoku, avšak často neobsahuje žádné předem vepsané nápovědy. Dodatečné informace k úspěšnému vyřešení poskytují menší označené celky různých tvarů s číslem, které uvádí součet číslic v daném útvaru. V jednom celku se součtem se nesmí číslice vyskytovat vícekrát.

3		15			22	4	16	15
25		17						
		9			8	20		
6	14			17			17	
	13		20					12
27		6			20	6		
				10			14	
	8	16			15			
				13			17	

Obrázek 2.10: Zadání sudoku typu „killer“. Převzato z Wikimedia [4]

Sudoku „větší-menší“ (angl. „Greater Than“) vychází z klasického sudoku, které je obohaceno o šipky místo hran uvnitř každého sektoru. Šipka napovídá vztah mezi sousedními čísly, ukazuje které je větší a které menší.



Obrázek 2.11: Zadání sudoku „větší-menší“. Převzato z Wikimedia [3]

KenKen vychází z latinského čtverce (na rozdíl od sudoku neobsahuje sektory). V zadání KenKen se nachází menší označené celky různých velikostí, ve kterých je uvedeno výsledné číslo a symbol matematické operace sčítání, odčítání, násobení nebo dělení. V každém tomto celku platí, že když na vyplněné číslice aplikujeme v libovolném pořadí danou operaci, získáme odpovídající výsledné číslo [16].

2.4.1 Další typy

Lze najít mnoho dalších zajímavých modifikací sudoku. Výše byly představeny pouze nejběžnější či jinak zajímavé verze sudoku.

2.5 Matematika v sudoku

Tato sekce se věnuje pouze klasickému sudoku, ne jeho alternativním verzím. Byla převzata z článku MIT Technology Review [2].

Existuje $6.670.903.752.021.072.936.960$ ($\doteq 6,7 \cdot 10^{21}$) způsobů, jak lze korektně vyplnit prázdnou sudoku mřížku. Nicméně mnoho z těchto možností jsou podobné, pouze vůči sobě otočené, osově či středově symetrické či mají prohozené umístění druhů číslic. Pokud vyloučíme tyto „duplicitní“ mřížky, zbyde pouhých 5.472.730.538 unikátních mřížek.

Sudoku vzniká vytvořením plně vyplněné mřížky a následným odebráním číslic do požadované obtížnosti. Správně vytvořené sudoku musí mít právě jedno řešení. Termín *minimální sudoku* značí ponechání nejmenšího počtu čísel z dané mřížky, při kterém má sudoku ještě stále pouze jedno řešení (odebrání libovolné další číslice by způsobilo vytvoření více možných řešení).

Nejmenší možný počet předem vyplněných číslic v sudoku je 17. Roku 2012 zveřejnil Gary McGuire studii, kterou dokázal, že neexistuje sudoku s 16 nápovědami, které by mělo jedno řešení. Jeho program ze všech 5.472.730.538 unikátních mřížek vytvořil minimální

sudoku a hrubou silou tuto hypotézu ověřil. Tento výpočet na tehdejších superpočítači v Irish Center for High-End Computing trval od ledna 2011 do prosince téhož roku.

Existující minimální sudoku s nejvíce nápovědami jich má 40. Nicméně toto číslo není podpořeno důkazem o neexistenci minimálního sudoku o více nápovědách a je tedy možné, že v budoucnu bude nalezeno minimální sudoku s vyšším počtem nápověd.⁵

2.6 Generování sudoku

Generování sudoku probíhá ve dvou krocích. Nejprve je vygenerována z náhodného základu zaplněná mřížka. Poté jsou z ní náhodně odebírána čísla jedno po druhém a v každém kroku je ověřeno, zda má sudoku pořád unikátní řešení (např. metodou backtracking). Při porušení této podmínky je třeba přidat zpátky poslední odebrané číslo pro vytvoření finální podoby zadání sudoku [11].

Čísla při vytváření zadání ze zaplněné mřížky nemusí být odebírána pouze náhodně. Často bývá zohledňována vizuální stránka vyplněných buněk. Sudoku může být středově symetrické (obrázek 2.12⁶), symetrické podle osy prostředního řádku/sloupce či diagonály, popř. jejich kombinace.

Čísla mohou být odebírána tak, aby zbylé číslice vytvořily obrázek či symbol. Například časopis Sudoku-K[15] vydávaný v Česku má na úvodní stránce sudoku, kde jsou nápovědy rozmístěny v tvaru dvojciferného čísla korespondujícím s číslem vydání. Lze také zvolit odebírání čísel z určitých pozic, aby sudoku vytvořilo neobvyklou výzvu (např. sektor bez jediné nápovědy, viz obrázek 2.13).

⁵U nejméně nápověd bylo pro ověření hrubou silou využito faktu, že do libovolného zadání sudoku můžeme vyplnit další číslici z jeho existujícího řešení a stále bude řešitelné s právě jedním řešením. Z toho vyplývá, že pokud neexistuje řešení pro sudoku s 16 nápovědami, nemohlo existovat ani vyřešitelné sudoku s 15 nápovědami atd. Pro důkaz neexistence sudoku s 16 nápovědami stačilo „pouze“ ověřit všechna minimální sudoku s 16 čísly. Pro maximální počet nápověd by však bylo potřeba ověřit všechny možné počty nápověd vyšší než 40.

⁶Tento obrázek byl vytvořen za pomoci aplikace popsané v této práci. Všechny obrázky v tomto vizuálním stylu byly vytvořeny stejným způsobem. Samotné zadání symetrického sudoku bylo taktéž vygenerováno mojí aplikací.

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

Obrázek 2.12: Ukázka vygenerovaného středově symetrického sudoku.

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

Obrázek 2.13: Ukázka vygenerovaného sudoku, které záměrně neobsahuje žádná čísla v prostředním sektoru ani v 5. řádku a 5. sloupci.

Pokud má vygenerované zadání sudoku být určité obtížnosti, kromě ověření jediného řešení sudoku je potřeba ohodnotit obtížnost. Je-li příliš vysoká, je nutné vrátit některé z odebraných čísel na místo a ověřit ji znova. Pokud má sudoku mít určitý tvar nebo je-li obtížnost příliš lehká, musí být vygenerována nová mřížka a postup zopakován.

Obtížnost samotná je subjektivní pojem. To, co je pro jednoho uživatele velmi těžké, může být pro druhého poměrně snadné. Stejně tak se vnímání obtížnosti liší mezi tvůrci zadání a programů, které je generují. Obtížnost je také přizpůsobena publiku. Sudoku označené jako „těžké“ publikované v magazínu s televizním programem bude mít jinou obtížnost než „těžké“ sudoku z magazínu věnujícímu se vytváření sudoku pro pokročilé luštitelé.

Kromě rozdílným hranic pro stupně obtížnosti se generátor od generátoru liší i samotný princip vyhodnocení složitosti. Ty nejjednodušší určují obtížnost pouze podle počtu vyplněných číslic v zadání. To může vést k zadání, které je označeno jako snadné, avšak pro jeho vyřešení by uživatel musel použít složitou strategii, a vzácně i naopak. I přes tyto nevýhody lze dle mé zkušenosti na takto vygenerovaná sudoku narazit např. V některých časopisech.

Kvalitnější generátory sudoku při přiřazení obtížnosti vychází z obtížnosti strategií k dosažení řešení. Kromě toho také často zohledňují další faktory, například rozložení číslic na mřížce⁷ a jejich počet.

⁷Sudoku s nerovnoměrně koncentrovanými nápovědami či s některými úplně prázdnými bloky bývá obvykle subjektivně obtížnější než sudoku s rovnoměrně rozmístěnými nápovědami.

Kapitola 3

Strategie řešení sudoku

Text této kapitoly, pokud nebude výslovně řečeno jinak, mluví o klasickém sudoku 9x9 s čtvercovými 3x3 sektory a pouze základními třemi pravidly – unikátnost čísla 1–9 v každém řádku, sloupci a sektoru. Většina těchto strategií však půjde použít i na ostatní typy sudoku s drobnými úpravami.

Postup řešení sudoku člověkem se skládá z opakovaných pokusů aplikovat strategie vedoucí k eliminaci více možností umístění číslice, čímž dochází k vyřešení sudoku.

Cílem u luštění sudoku je doplnit do hrací plochy všechna čísla tak, aby odpovídala zadaným pravidlům. V žádném okamžiku postupu řešení by řešitel neměl hádat umístění číslice¹. Správné zadání sudoku má právě jedno správné řešení.

Postup řešení zahrnuje vyplnění volných políček všemi číslicemi, tzv. kandidáty, a následně odmazávání kandidátů v rozporu z pravidly pro vyplnění sudoku, dokud v políčku nezbyde pouze jedna číslice. Cílem každé strategie je vyloučit umístění alespoň jedné kandidátní číslice.

U snadných sudoku obvykle uživatel nepotřebuje vyhledávat složité postupy a dostane se k řešení za pomoci snadno aplikovatelných postupů². Obtížná sudoku vyžadují aplikaci komplikovanějších postupů.

Složitost strategie je subjektivní záležitost. Strategii činí obtížnou dva faktory – schopnost vidět výchozí podmínky pro aplikaci strategie a schopnost strategii aplikovat. V této sekci jsou strategie seřazeny vzestupně podle subjektivní obtížnosti.

Strategie pro řešení sudoku nepodléhají žádnému oficiálnímu standardu. Neexistuje žádný oficiální soubor pravidel pro řešení sudoku, který by určoval náročnost jednotlivých strategií a jejich název. Je pravděpodobné, že některé zde uvedené strategie mohou být v jiných zdrojích uváděny s jiným názvem či jiným ohodnocením složitosti.

Tato práce se nesnaží shrnout všechny možné strategie ani objevit další. Jejím cílem je posbírat základní a obecně rozšířené strategie a představit je uživateli formou, která povede k jeho zdokonalení se v řešení sudoku.

Názvy strategií se v anglickém jazyce ustálily na zaběhnutých výrazech. V českém jazyce ovšem jejich alternativy běžné nejsou. Většina zdrojů je nepřekládá a místo toho je používá v původní formě. V této práci taktéž zachovávám anglické názvy známých strategií. V používaném názvosloví vycházím z vícero zdrojů [5][9][18][14].

¹Samozřejmě existují zadání sudoku, kde k řešení nelze dospět jinak než hádáním umístění číslic. V této práci se však zabývám logickými cestami k vyřešení sudoku a tedy pouze těmi zadáními, které lze vyřešit bez hádání.

²Mnoho ze snadných strategií může řešiteli připadat „očividných“ a často si tak nemusí být vědom, že právě aplikoval nějakou strategii.

Při popisu strategií vycházím především z mé vlastní zkušenosti se strategiemi, které jsem si za roky luštění sudoku osvojila. Zmíněné zdroje byly použity na sjednocení názvů a na kontrolu logických aspektů strategií.

3.1 Odstranění přímých kolizí

Před aplikací první strategie je vhodné odstranit kandidáty z těch bloků, ve kterých je již daná číslice vyplněná. Tento krok je vhodné provést pokaždé, kdy se díky některé strategii vyplní další číslice.

3.1.1 Změny pro alternativní typy sudoku

U klasického sudoku je za blok označován každý sloupec, řádek a sektor. Některé typy sudoku definují kromě sloupců, řádků a sektorů také další sadu buněk, ve kterých se musí dodržet podmínka o umístění všech čísel právě jednou. U těchto sudoku jsou pod pojmem blok chápány také tyto přidané sady buněk. Např. blokem diagonálního sudoku může být řádek, sloupec, sektor nebo diagonála.

Když strategie mluví o aplikování určitého postupu na blok, u speciálních sudoku to zahrnuje i přidané sady buněk. U následujících strategií už to nebude zmíněno, ale platí to i v nich.³

3.2 Naked single

V jednom nevyplněném políčku se vyskytuje pouze jeden kandidát. Můžeme do pole doplnit dané číslo.

3.3 Hidden single

V rámci jednoho bloku se vyskytuje kandidát pouze v jedné buňce. V takovém případě je možné ho do ní napsat.

3.4 Naked Pairs, Triples and Quads

Naked Pairs hledá dvojici políček v rámci jednoho bloku, které obsahují pouze tytéž dva kandidáty (*pair*). Je zřejmé, že ve výsledném sudoku bude buď první kandidát v prvním políčku, čímž na druhé políčko zbyde jako jediná možnost druhý kandidát, nebo budou na prohozených pozicích. Nevíme zatím, která z možností nastane, ale můžeme jednoznačně určit, že oba kandidáty v rámci bloku budou zde a nemohou být na ostatních políčkách v daném bloku.

³V popisech strategií bude poměrně málo úprav pro alternativní typy sudoku. To je způsobeno především tím, že pokud na sady buněk spadající do nových pravidel nahlížíme jako na blok, dají se strategie generalizovat.

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

Obrázek 3.1: Nalezený *Naked pair* označený žlutě vylučuje umístění kandidátů 1 a 2 v ostatních políčkách řádku (označeno červeně). (Zobrazena pouze relevantní část sudoku.)

Naked Triples je založen na stejném principu jako *Naked Pair* – hledá trojici políček obsahující pouze trojici kandidátů. *Naked Quad* hledá čtveřici políček se čtveřicí kandidátů.

Změny pro alternativní typy sudoku

Strategie „Hidden“ a „Naked“ se navzájem doplňují. Platí, že nachází-li se v bloku naked X , kde X je maximální velikost množiny kandidátních čísel nacházejících se v X buňkách bloku, nachází se tam také hidden $N - X$, kde N je počet symbolů sudoku.

Pro sudoku 9x9 stačí kvůli této vlastnosti hledat dvojice až čtveřice. V sudoku 16x16 je třeba hledat dvojice až osmice. V sudoku 4x4 stačí hledat pouze dvojice.

	1	2	3	4	5	6	7	8	9
1	A	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6					
2	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6					
3	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G					
4	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G	1 2 3 4 5 6 7 8 9 C D E F G	B				
5									

Obrázek 3.2: Sudoku 16x16. Nalezené *Naked 6* označené žlutě vylučuje umístění kandidátů 1–6 v ostatních políčkách sektoru (označeno červeně). (Zobrazena pouze relevantní část sudoku.)

3.5 Hidden Pairs, Triples and Quads

Hidden Pairs hledá dvojici kandidátů, které se dohromady vyskytují pouze ve dvou políčkách z celého bloku. Pokud se dva kandidáty vyskytují pouze na dvou pozicích, ve výsledném řešení budou určitě společně zabírat tyto pozice a lze z těchto políček odstranit ostatní kandidáty.

Hidden Triples hledá trojici kandidátů vyskytujících se pouze na třech políčkách, *Hidden Quad* hledá čtveřici na čtyřech políčkách.

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

Obrázek 3.3: Nalezený *Hidden Pair* označený žlutě vylučuje umístění ostatních kandidátů v jeho buňkách (označeno červeně). *Hidden N* strategie a *Naked N* se vyskytují zároveň, zde je vidět *Naked Triple* (buňky s černými kandidáty). (Zobrazena pouze relevantní část sudoku.)

3.5.1 Změny pro alternativní sudoku

U sudoku velikosti 4x4 není *Hidden Pair* vůbec třeba hledat. Pokud by se tam vyskytoval, tedy pokud by dva druhy čísel byly pouze ve dvou buňkách bloku, ve zbylých buňkách bloku by zbyly maximálně dva kandidáty a aplikovala by se některá z předchozích strategií.

Pro sudoku 6x6 má ze stejného důvodu cenu hledat pouze *Hidden Pair*, ale už ne *Hidden Triple/Quad*.

V sudoku 16x16 se vyplatí hledat *Hidden 2–8*.

3.6 Intersection Removal

Tato strategie bývá označována také jako *Locked Pairs/Triples*. Zkoumá vztah kandidátů na průsečíku čtvercového sektoru a sloupce či řádku. Průsečík tvoří 3 buňky.

Strategie zkoumá dvě možnosti:

- Průsečík obsahuje všechny výskyty některého kandidátu v rámci sektoru. Viz obrázek 3.4.

Aby byla splněna podmínka výskytu čísla v sektoru, je jisté, že se musí vyskytovat na jedné ze tří (popř. dvou) pozic na průsečíku. Tím je vyloučeno, aby se kandidát nacházel na jiné pozici v řádku/sloupce mimo tuto intersekcii.

- Průsečík obsahuje všechny výskyty některého kandidátu v rámci řádku/sloupce. Viz obrázek 3.5.

Je jisté, že se v průsečíku musí kandidát nacházet na jedné ze tří (popř. dvou) pozic, aby byl zachován jeho výskyt v daném řádku/sloupce. Proto se nemůže nacházet jinde v aktuálním sektoru kromě těchto pozic v průsečíku.

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

Obrázek 3.4: (Zobrazena pouze relevantní část sudoku.) Ukázka použití strategie *Intersection Removal* ze sektoru do řádku. V sektoru se všechny výskyty čísla 1 nachází v prostředním řádku (žlutě). Můžeme proto v druhém řádku mimo tento sektor vymazat ostatní výskyty kandidátu 1 (červeně).

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

Obrázek 3.5: (Zobrazena pouze relevantní část sudoku.) Ukázka použití strategie *Intersection Removal* z řádku do sektoru. V prvním řádku se kandidát 2 vyskytuje pouze v rámci jednoho sektoru (žlutá). Aby byl splněn jeho výskyt v řádku, nemůže se vyskytovat v sektoru na jiných pozicích než v rámci zkoumaného řádku a můžeme kandidáty v druhém a třetím řádku tohoto sektoru vymazat (červená).

3.6.1 Změny pro alternativní typy sudoku

V typech sudoku se sektory jiného než čtvercového tvaru⁴ se strategie *Intersection removal* dá aplikovat ještě třetím způsobem.

Libovolná buňka $rXcY$ ležící mimo zkoumaný nečtvercový sektor se může nacházet ve stejném řádku či sloupci jako několik buněk zkoumaného sektoru. Pokud se v rámci daného sektoru nachází určité kandidátní číslo pouze v takto sdílených buňkách, nemůže se toto číslo nacházet ve vybrané buňce $rXcY$ mimo sektor.

⁴Sektor(y) jiného tvaru mají sudoku: Diagonální (dva sektory diagonál), sudoku středy čtverců (sektor středů čtverců) a jigsaw sudoku (9 nepravidelných sektorů).

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

Obrázek 3.6: Ukázka použití speciálního případu strategie *Intersection Removal* v diagonálním sudoku. Na diagonále je možné umístit číslici 1 pouze na dvě pozice (žlutě). Protože obě možné umístění vylučují umístění číslice 1 v r2c4 a r4c2, mohou odtud být kandidátní čísla odstraněna (červeně). (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

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

Obrázek 3.7: Ukázka použití speciálního případu strategie *Intersection Removal* v sudoku jigsaw. V rámci vyplněného sektoru (na řádcích 6 až 8) se číslo 1 může nacházet na třech pozicích (žlutě). Každá z těchto pozic vylučuje umístění 1 v r6c2, a proto odtud může být kandidátní číslo 1 odstraněno (červeně). (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

3.7 X-Wing

Pokud se ve dvou různých řádcích sudoku nachází kandidátní číslo pouze na dvou pozicích, které leží ve stejných sloupcích jako totéž kandidátní číslo v druhém řádku, lze aplikovat strategii *X-Wing*.

Existují pouze dvě možnosti, jak budou kandidáty v obou řádcích umístěny, a každá z nich vyloučí umístění tohoto kandidátu v sloupci mimo tyto řádky.

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

Obrázek 3.8: Ukázka použití strategie *X-Wing*. V řádku 3 a 5 se kandidátní číslo 1 nachází pouze ve dvou buňkách, a tyto buňky leží ve dvou stejných sloupcích (žlutě). Číslo 1 proto v těchto sloupcích musí být v jednom z vybraných řádků a můžeme ji ze zbylých pozic odstranit (červeně). (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

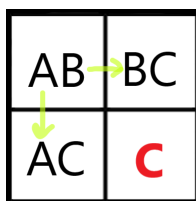
Název strategie vznikl z vizuální podoby nalezených dvojic kandidátů. Když se spojí dvě možné umístění číslice, které mohou nastat, tedy protilehlé rohy čtverce, vytvoří písmeno X.

Vysvětlení a obrázek 3.8 popisují nalezení podmínek pro strategii *X-Wing* díky omezení řádku a promítnutí strategie do sloupců. Strategie může fungovat také naopak, omezení tvořící *X-Wing* může být nalezeno ve sloupci a promítnuto do řádku.

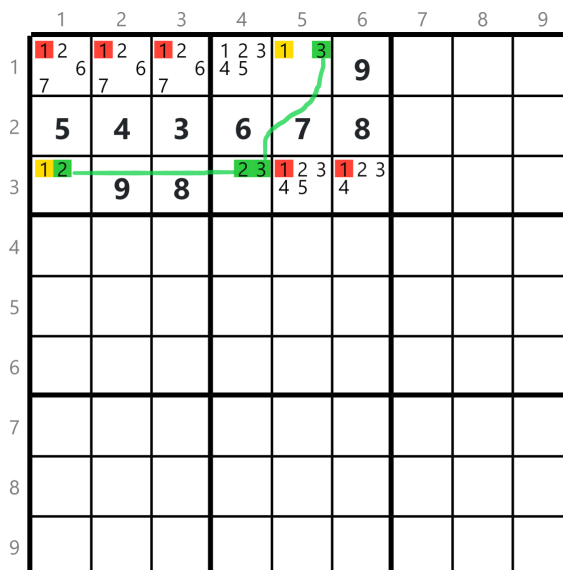
3.8 Y-Wing

Strategie *Y-Wing* je krátkou verzí *XY-Chain*, která spojuje pouze 3 buňky s dvojicí kandidátů.

Strategie je symbolicky s pomocí písmen místo číslic znázorněna na obrázku 3.9. Písmena A, B a C zastupují libovolné číslo daného druhu sudoku. Buňka s AB je ve stejném bloku (sloupci) jako buňka s AC a ve stejném bloku (řádku) jako buňka s BC. Ať by bylo do buňky AB doplněno a nebo B, v jedné z buněk, které ovlivňuje, by zůstalo jako jediná možnost C. Proto není možné, aby bylo v levé dolní buňce C umístěno a tento kandidát strategie odstraní.

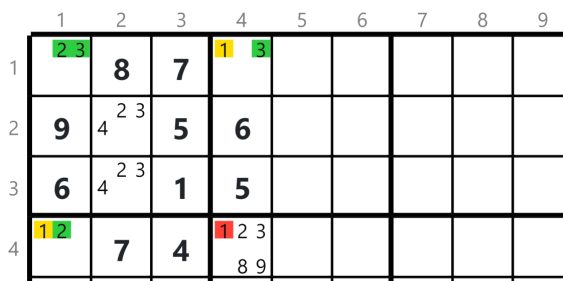


Obrázek 3.9: Symbolické znázornění strategie *Y-Wing*, symboly ABC zastupují libovolné číslo.



Obrázek 3.10: *Y-Wing* vychází z buňky r3c4. Tato buňka ovlivňuje další dvě buňky o dvou kandidátech (zelené linky), se kterými se nachází v jednom bloku. Ať bude v r3c4 vyplněna 2 nebo 3, v jedné z navázaných buněk kvůli tomu zůstane pouze kandidát 1. Můžeme proto odstranit 1 ze všech buněk (červeně), které jsou v témže bloku s r3c1 i r1c5. (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

Svůj název strategie získala díky možnému rozmístění buněk, které ji tvoří (viz obrázek 3.11).



Obrázek 3.11: Umístění *Y-Wing*, podle kterého strategie získala jméno. Umístění zeleně označených čísel s trochou fantazie po vzájemném propojení písmeno Y. (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

3.9 Swordfish

Strategie *Swordfish* je založená na stejném principu jako X-Wing, avšak pracuje s výskytem 3 kandidátů.

Pokud se jeden kandidát nachází v třech řádcích pouze na třech pozicích a všechny tyto pozice leží pouze ve třech sloupcích (viz obrázek 3.12), lze aplikovat tuto strategii. Kandidátní čísla jsou omezena řádky natolik, že při libovolném umístění bude v každém sloupci vyplněno jedno v rámci těchto řádků – kandidát se tedy nemůže nacházet na ostatních pozicích sloupce.

Tato strategie může také vyjít z omezených sloupců, které se promítnou do řádků.

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

Obrázek 3.12: Ukázka použití strategie *Swordfish*. V řádku 2, 4 a 7 se kandidátní číslo 1 nachází pouze ve třech buňkách (žlutě), a tyto buňky leží ve třech stejných sloupcích 3, 5 a 7. Číslo 1 proto v těchto sloupcích musí být v jednom z vybraných řádků a můžeme je ze zbylých pozic odstranit (červeně). (Zobrazena pouze část čísel relevantních k demonstraci strategie.)

3.10 XY-Chain

Strategie *XY-Chain* se zakládá na stejném principu jako Y-Wing, avšak není omezena pouze na dvě navzájem spojené dvojice. Své jméno získala díky tomu, že se v ní řetězí buňky.

Nalezený XY-Chain se skládá z libovolného počtu buněk, které obsahují pouze dvě kandidátní čísla. Platí, že dvě spojené buňky se nachází v jednom bloku a mají jedno z těchto kandidátních čísel stejné (toto číslo je spojuje). Zároveň musí platit, že pokud buňka není na kraji řetězce, musí být s každým ze svých dvou spojených buněk propojena rozdílným kandidátním číslem. Uvažujme následující řetězec:

A B	B C	C D	D E	E A
-----	-----	-----	-----	-----

Buňky nemusí ležet všechny ve stejném bloku, pouze dvě spojené musí sdílet blok (AB a BC musí ležet v jednom bloku, BC zároveň ve stejném jako CD, apod.).

Pokud se na koncích takového řetězce nachází stejné nespojené číslo, v tomto případě A, je jisté, že na jedné z těchto pozic se a musí nacházet. Pokud existuje ve zbytku sudoku buňka s kandidátem A, která „vidí“ (nachází se ve stejném bloku jako ony) tyto konce řetězce s A, můžeme z ní a odstranit.

Pro mentální kontrolu je dobré představit si, jak by byly buňky doplněny, kdyby na jednom z konců řetězce bylo vyplněno číslo, kterým začíná první spojení. Pokud do první buňky bude doplněno B, v druhé musí být C atd. A pro poslední buňku zbyde jediná možnost – A. Při kontrole z druhé strany by měl vyjít stejný výsledek, kterým se můžeme ujistit, že je řetězec skutečně XY-Chain⁵.

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

Obrázek 3.13: Zeleně jsou označeny kandidáty tvořící řetězec, na jehož koncích je pokaždé 1 (žlutě). Protože se jedná o XY-Chain, bude se kandidát 1 nacházet právě na jedné ze žlutě označených pozic. Červeně označená kandidátní čísla se nachází ve stejném bloku s oběma konci XY-Chain, a je tedy možné je odstranit. (Zobrazena pouze část sudoku relevantní pro použití strategie.)

⁵U této strategie, obzvláště pokud je spojený řetěz hodně dlouhý, je snadné se přehlédnout a chybně napojit dvě buňky za sebe stejným kandidátem. Tato mentální kontrola ověří, že buňky se navzájem skutečně ovlivňují žadáním způsobem.

Kapitola 4

Návrh aplikace

Nástroj bude vyvíjen jako webová stránka. Hlavním důvodem pro toto rozhodnutí byla dostupnost programu pro uživatele. Předpokládám, že průměrný uživatel na stránku zavítá na pár minut, aby zadal svoje sudoku a zjistil nápovědu pro další krok, popř. aby se naučil strategii. Webová stránka bude sloužit jako nápověda „po ruce“, kterou uživatel může a nemusí využít. Nutnost stáhnout si aplikaci (ať už na Windows nebo chytrý mobil) by značně omezilo pohodlí uživatele a dosah nástroje.

Uživatelské rozhraní aplikace vychází z možných uživatelských potřeb pro řešení sudoku a vzdělávání se v této oblasti. Cílem je poskytnout pohodlné prostředí pro luštění sudoku a nabídnout pomoc, která uživateli ukáže, jak je možné nad sudoku přemýšlet.

Uživatel bude moci využít aplikaci pouze pro informování se o způsobu strategie, nahrát do aplikace vzorové sudoku a v rámci webové stránky ho pohodlně luštit nebo si ho stáhnout pro tisk, či zadat svoje rozluštěné sudoku z časopisu a vyhledat si další krok.

4.1 Omezení aplikace

Jak bylo nastíněno v předchozích kapitolách, problematika sudoku je velmi rozsáhlá.

Lze zkoumat generování sudoku a ohodnocování jeho obtížnosti. Na řešení sudoku se lze dívat jako na problém prohledávání stavového prostoru, nebo se naopak dá zaměřit na řešení sudoku logickou cestou pomocí strategií. Tyto strategie je možné zkoumat a hledat nové strategie pro složitější zadání sudoku či hledat způsob úpravy stávajících, aby byly snadněji zpozorovatelné.

Sudoku existuje v mnoha podobách a tvarech. Přidáním dalších omezení a pravidel, popř. využitím dalších principů jako matematických operací, lze vytvořit další a další druhy a pro každý takový druh je třeba upravit generování, strategie i způsob řešení.

Další zajímavou oblastí spojenou se sudoku je počítačové vidění, které lze využít na převedení vyfoceného zadání do počítačové podoby.

V neposlední řadě je také možné zkoumat sudoku ve vztahu k uživateli. Pokud se na sudoku rozhodneme dívat jako na zábavu pro uživatele, je třeba si klást otázky, jakou aplikaci by pro svoje luštění potřeboval, co už existuje a co naopak není snadné najít, a jaké možnosti by mu pomohly se v sudoku zdokonalit.

Z praktických důvodů tato práce nemůže postihnout všechny možné vlastnosti, které může aplikace zabývající se sudoku mít.

Rozhodla jsem se proto soustředit pouze na následující aspekty:

- Pohodlné prostředí pro řešení sudoku online.

- Představení různých typů sudoku a poskytnutí vzorových zadání k nim.
- Vysvětlení a demonstrování strategií řešení sudoku¹.
- Získání nápovědy pro další krok z pohledu luštitel.
- Ověření řešitelnosti logickou cestou.

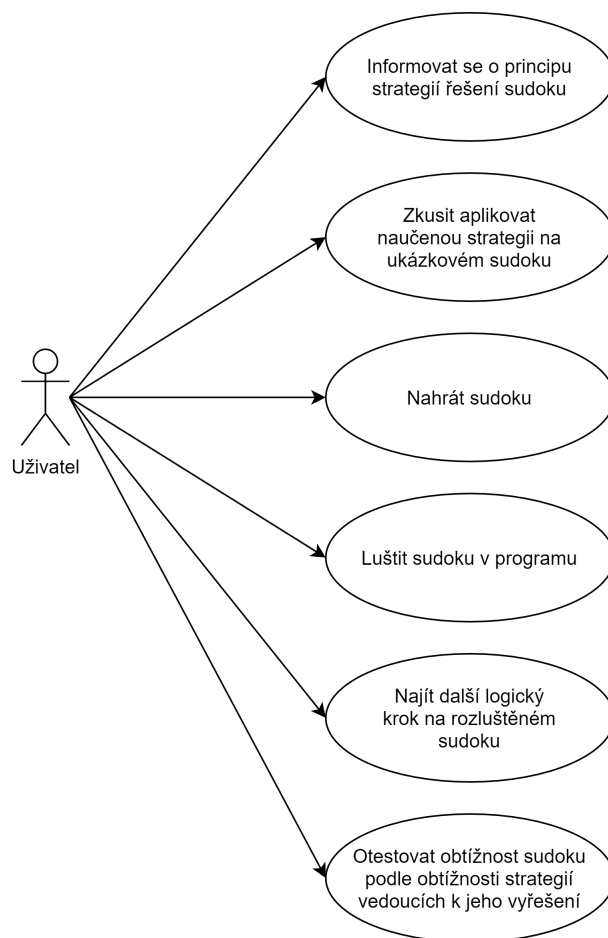
A výše zmíněný seznam v plném rozsahu aplikovat pro následující typy sudoku:

- Sudoku 9x9
- Sudoku 4x4
- Sudoku 6x6
- Sudoku 16x16
- Diagonální sudoku 9x9
- Sudoku středy čtverců 9x9
- Diagonální sudoku se středy čtverců 9x9
- Hypersudoku 9x9
- Jigsaw 9x9 s uživatelem zadanými tvary sektorů

4.2 Diagram případů užití

Na obrázku 4.1 jsou znázorněny případy užití, ze kterých vychází návrh aplikace.

¹Zahrnu všechny z kapitoly 3. Existuje samozřejmě i mnoho složitějších strategií, avšak tuto oblast přenechám pro experty na řešení sudoku. Na sudoku, kde by byly takové strategie potřeba, běžný uživatel ani pokročilý luštitel téměř nenarazí.



Obrázek 4.1: Diagram případů užití navrhované aplikace.

Jediným aktorem je uživatel, který na stránku zavítal. Aplikace neuchovává žádná persistentní data, není potřeba role administrátora. Jediné dočasně uchovávané informace se týkají nastavení ovládání programu uživatelem a budou uchovávány pomocí session.

Podrobnější popis jednotlivých akcí uživatele:

- *Informovat se o principu strategií řešení sudoku:* Uživatel bude mít možnost najít v nástroji vysvětlení každé strategie z kapitoly 3. Strategie bude obsahovat popis výchozích podmínek pro aplikaci strategie a efekt použití strategie. Bude vysvětlena s použitím obrázků přímo z nástroje, aby se uživatel lépe orientoval v použitém grafickém zvyraznění.
- *Zkusit aplikovat naučenou strategii na ukázkovém sudoku:* Popis strategie bude navíc obsahovat odkaz, který uživatele přesune na stránku řešení sudoku a nahraje do mřížky připravené sudoku, ve kterém je pro doplnění další číslice třeba použít danou strategii. Tato možnost dá uživateli prostor aplikovat naučené znalosti.²

²Učení nové strategie může probíhat ve třech krocích.

V prvním kroku se seznámí s teoretickou stránkou strategie a pochopí její princip na přiloženém příkladu.

V druhém kroku se pokusí aplikovat strategii na připraveném sudoku. V tomto kroku již uživatel musí být schopen rozpoznat výchozí podmínky pro aplikaci strategie, ale má výhodu znalosti, že hledá právě tuto

- *Nahrát sudoku:* Nástroj by nedával smysl bez možnosti zadat výchozí stav sudoku. Zadání sudoku by mělo být možné pohodlně a vícero způsoby podle uživatelské potřeby. Bude možné nahrát jedním kliknutím připravené ukázky sudoku. Uživatel bude moci zadat svoje vlastní zadání pomocí fyzické či virtuální klávesnice a toto zadávání bude usnadněno pomocí klávesových zkratk, které usnadní pohyb mezi buňkami pro zadání čísel. Kromě toho bude také možné nahrát sudoku z externích zdrojů přes specifický odkaz (jak bude popsáno podrobněji v sekci 4.6).
- *Luštit sudoku v programu:* Nástroj bude klást velký důraz na pohodlí luštění sudoku. Většina nástrojů na trhu se zaměřuje buď pouze na hledání řešení nebo jen na prostor pro řešení sudoku, a uživatel je tak ochuzen o možnost řešit sudoku vlastními silami a dotázat se v případě potřeby na pomoc s nalezením dalšího kroku.

Vyvíjený nástroj bude obsahovat funkce pro komfortní luštění jako je automatické doplnění kandidátních čísel do volných políček a zvýraznění umístění jedné číslice v celém sudoku. Dále bude možné využít klávesových zkratk pro pohyb mezi buňkami sudoku a bude mít možnost nastavení některých ovládacích prvků.

- *Najít další logický krok na rozluštěném sudoku:* Jádrem celého nástroje je nalezení dalšího logického kroku v řešení zadaného sudoku.

Po stisku odpovídajícího tlačítka se nástroj pokusí najít nejlehčí strategii, která by v právě zadaném sudoku vyplnila číslici či odstranila alespoň jedno kandidátní číslo. Pokud je sudoku neřešitelné, obsahuje prázdné buňky bez možných kandidátních čísel, má mezi vyplněnými buňkami čísla v přímé kolizi, nebo už je plně vyřešené, uživatel je o takové skutečnosti informován.

Pokud nástroj našel aplikovatelnou strategii, označí ji v seznamu strategií. Tento seznam je seřazen od nejlehčí po nejtěžší a slouží jako vizuální reprezentace pořadí, ve kterém jsou strategie v sudoku vyhledávány. Zároveň slouží jako odkaz na vlastní stránku strategie s podrobným popisem. Nalezená strategie se graficky vyznačí v sudoku a její princip bude popsán pod mřížku.

Uživatel bude moci tento krok potvrdit, čímž se aplikuje (vyplní se čísla do buňky či odstraní kandidátní čísla), nebo bude možné navrhovaný krok zrušit. Zrušení odstraní grafické zvýraznění a popis a umožní uživateli zkusit aplikovat strategii samostatně.

- *Otestovat obtížnost sudoku podle obtížnosti strategií vedoucích k jeho vyřešení:* Nástroj si okopíruje rozluštěné sudoku (uživateli se rozluštěné sudoku nezmění) a pokusí se ho postupným hledáním dalších logických kroků vyřešit. Pokud se mu to nepovede, vrátí uživateli tuto informaci a bližší vysvětlení. Pokud sudoku vyřeší, oznámí to uživateli a uvede, jakou nejtěžší strategii k tomu musel použít.

Informace o nejtěžší strategii slouží jako odhad obtížnosti celého sudoku.

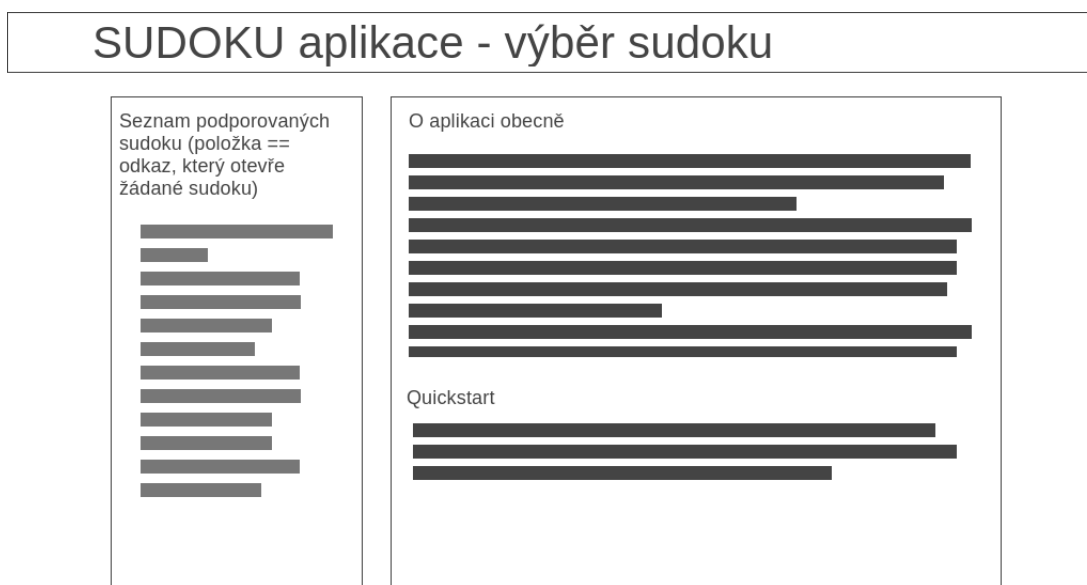
strategii. Posledním krokem je schopnost použít strategii na sudoku, ve kterém nevěděl, kterou strategii musí použít. Pokud na takovém sudoku dokázal úspěšně rozpoznat a aplikovat korektní postup, zvládl i třetí krok.

Je důležité, aby správný výukový program dal uživateli prostor vyzkoušet si samostatně aplikovat strategii na sudoku připraveném pro její použití.

4.3 Grafické uživatelské rozhraní

4.3.1 Řešení sudoku

Pro výběr typu sudoku se uživateli zobrazí stránka znázorněná na obrázku 4.2. V levém sloupci uvidí seřazené typy sudoku s odkazy na stránku, kde bude možné ono sudoku řešit. V pravé části bude pro nové uživatele popis, jak začít, a budou tam blíže popsány jednotlivé druhy sudoku.

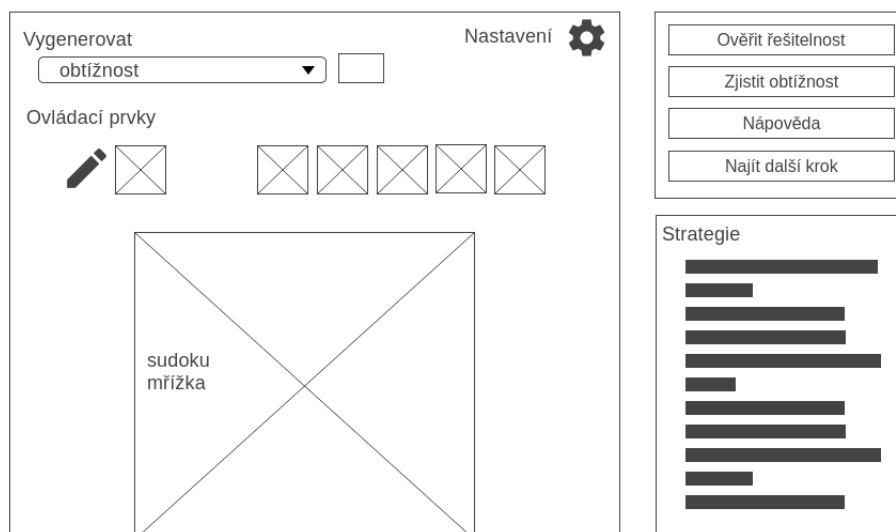


Obrázek 4.2: Výchozí stránka s výběrem typu sudoku.

Po výběru sudoku se uživateli zobrazí stránka znázorněná na obrázku 4.3³. V levé části se budou nacházet hlavní ovládací prvky sudoku a samotná mřížka sudoku. Pravý sloupec bude obsahovat prvky nápovědy při luštění, především je zde umístěna možnost najít další logický krok a seznam strategií k tomu využitých.

³Výsledná aplikace se oproti tomuto návrhu mírně liší. Možnost vygenerovat sudoku byla změněna na nahrání sudoku z předem daného seznamu zadání. Důvod této změny je popsán v sekci 5.2. Tlačítko „Nápověda“ bylo z výsledné implementace odstraněno, protože jeho zamýšlená funkce byla téměř totožná s funkcí „Najít další krok“. Tlačítka „Ověřit řešitelnost“ a „Zjistit obtížnost“ byly sjednoceny do jednoho tlačítka „Otestovat řešitelnost a obtížnost“, které slučuje jejich funkce.

SUDOKU aplikace - vybraný typ sudoku



Obrázek 4.3: První návrh stránky s konkrétním typem sudoku obsahující interaktivní prvky, díky kterým může uživatel vygenerovat sudoku, získat nápovědu, vyhledat konkrétní strategii, vyplnit sudoku či zvýraznit čísla pro přehlednost.

4.3.2 Jigsaw

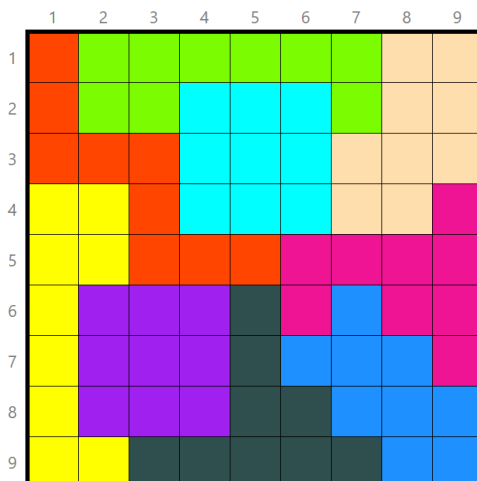
Sudoku Jigsaw může mít mnoho podob. Rozložení jeho sektorů má vliv nejen na grafické zvýraznění pro uživatele, ale i na hledání dalšího kroku – program na straně serveru zpracovávající požadavek na další krok musí mít informaci o aktuálním rozložení sektorů.

Nelze předem zjistit, jaké možné verze rozložení sektorů může uživatel chtít zadat s požadavkem na vyřešení zadání. Proto musí nástroj obsahovat možnost zadat vlastní rozložení sektorů.

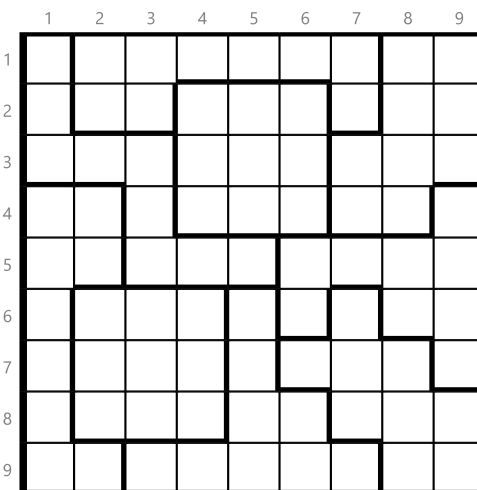
Při prvním spuštění otevření řešení sudoku jigsaw, kdy uživatel nemá v session připojení uložené vlastní rozmístění, musí být uživatel přesměrován na stránku, kde si bude moct sektory definovat. Na tuto stránku se poté bude moct kdykoliv vrátit a redefinovat rozmístění sektorů.

Uživateli se na stránce zobrazí prázdná sudoku mřížka a bude vyzván, aby postupně vybral pro každý sektor 9 políček. Každému sektoru bude přiřazená barva pro přehlednou grafickou reprezentaci a uživatel bude informován o tom, kolik políček daných sektorů ještě zbývá označit. Pro vytváření sektorů bude platit, že buňky každého z nich musí tvořit souvislou plochu o právě 9 buňkách. Na obrázku 4.4 se nachází platné rozmístění sektorů zadané uživatelem.

Po odeslání nového rozmístění sektorů se tato informace uloží do session a bude používána pro vykreslení sudoku mřížky (obrázek 4.5) a pro výpočet dalších kroků, dokud uživatel nevytvoří nové.



Obrázek 4.4: Vytváření nových sektorů jigsaw pomocí barev.



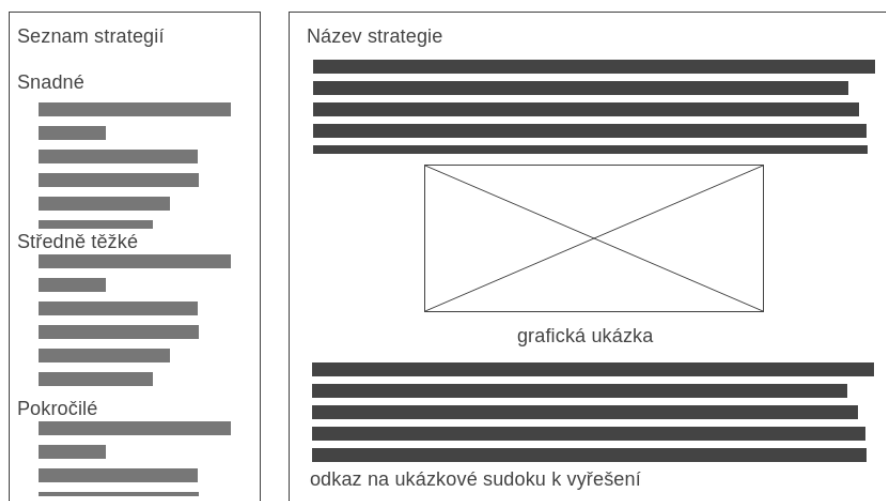
Obrázek 4.5: Výsledná mřížka sudoku jigsaw s vlastními tvary sektorů.

4.3.3 Návod

Cesta vysvětlení strategie uživateli se bude zakládat na třech principech – vysvětlit teorii strategie, názorně ji ukázat a dovolit uživateli, aby si ji sám vyzkoušel aplikovat.

System bude obsahovat stránku pro každou strategii z kapitoly 3. Jak je znázorněno na obrázku 4.6, na stránce bude strategie podrobně popsána, její rozpoznání a aplikace budou ukázána na příkladech a bude rovněž přiložen odkaz na rozpracované sudoku, kde bude právě tuto strategii potřeba použít k jeho vyřešení.

SUDOKU aplikace



Obrázek 4.6: Návrh rozložení stránky s popisem konkrétní strategie.

Všechny návody budou obsaženy ve vlastní sekci webové stránky. Důležité je, aby uživatel byl schopen pohodlně najít strategii, která ho zajímá. Vedle řešeného sudoku bude proto umístěn také seznam strategií pro aktuální sudoku, kde uživatel uvidí při vyhledání dalšího kroku právě aplikovanou strategii. Kliknutím na tyto názvy v seznamu bude uživatel odkázán na její podrobný popis (viz obrázek 4.3, kde se seznam strategií nachází v pravém dolním rohu stránky).

4.4 Formát sudoku

Aplikace je navržena tak, aby nemusela uchovávat aktuální stav sudoku na straně serveru. V průběhu luštění bude uživatel vpisovat číslíce – aktuální stav sudoku – pouze do html podoby načtené stránky. Sudoku nebude uloženo a v případě obnovení stránky dojde ke ztrátě postupu.

Při zavolání funkce na nalezení dalšího kroku či ověření obtížnosti bude vytvořen JSON obsahující informace o aktuálním stavu sudoku v prohlížeči, tj. stav každé z buněk sudoku a informace o číslech a kandidátních číslech, které obsahuje.

Na straně serveru bude z tohoto JSONu vytvořena třída reprezentující stav sudoku. Při nalezení aplikovatelné strategie server klientovi nepředává celé sudoku, ale pouze změny, které strategie vykoná.

4.5 Řešení sudoku počítačem

Server bude obsahovat třídu starající se o aplikování strategií. Tato třída bude obsahovat informace o aktuálním typu sudoku, pomocné proměnné pro rychlejší vyhledávání v sudoku (např. mapování každé buňky na číslo řádku, sloupce a sektoru, ve kterém se nachází) a implementované metody odpovídající jednotlivým strategiím popsáním v kapitole 3. Všechny

strategie budou převedeny do podoby strojově aplikovatelného algoritmu pracujícím nad třídou tvořící formát sudoku.

Aplikování strategií bude potřeba pro dvě situace:

- *Uživatel chce nalézt další logický krok.*

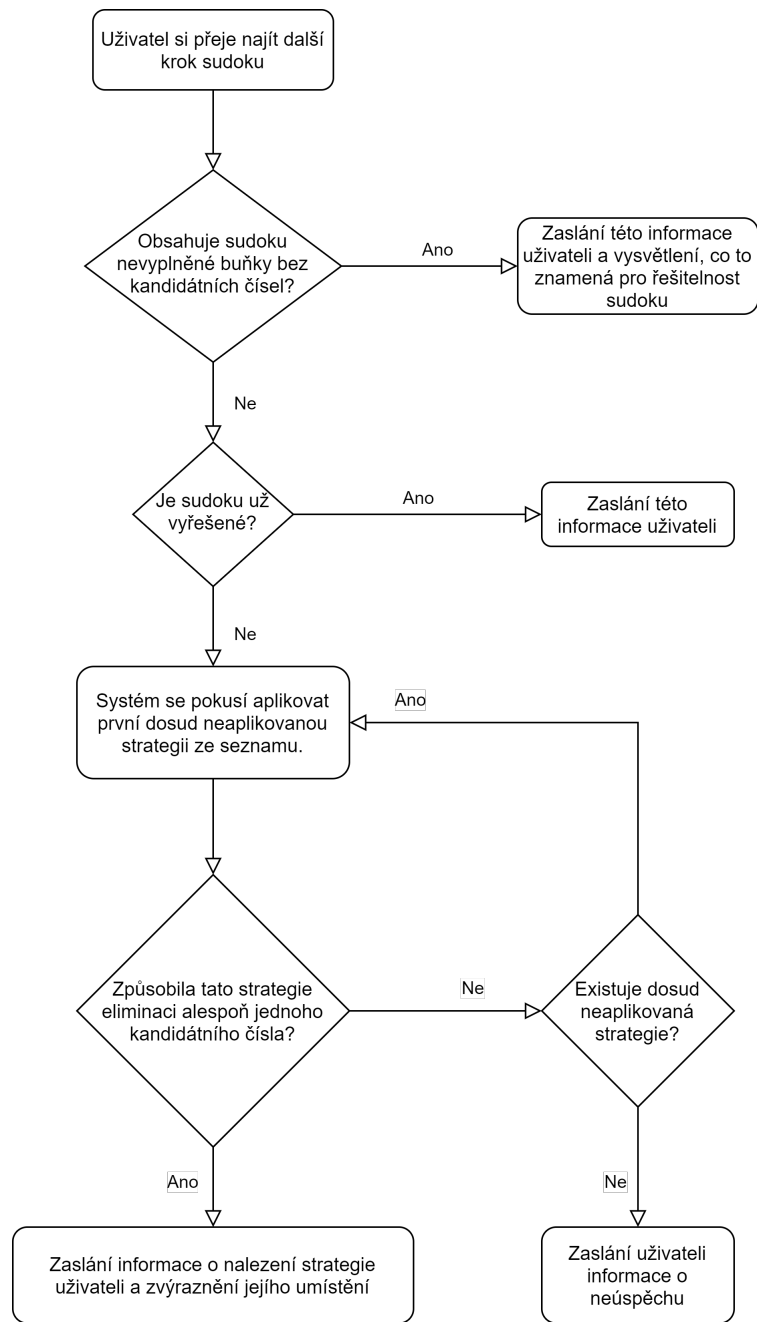
Na vývojovém diagramu 4.7 je znázorněn postup aplikace pro nalezení dalšího logického kroku. Nejprve dojde k ověření, zda sudoku není již vyřešené či zda není kvůli některé vlastnosti neřešitelné. Pokud sudoku takovou vlastnost nemá, je postupně na sudoku vyzkoušena aplikace každé implementované strategie. Pokud tato strategie způsobí vyplnění čísla či odstranění kandidátního čísla, je považována za úspěšnou a uživateli je vrácena informace o jejím použití. Pokud by došlo k vyčerpání strategií bez efektu na sudoku, uživatel je informován o selhání logického způsobu řešení⁴.

- *Uživatel si přeje zjistit, zda je sudoku řešitelné logickou cestou a jakou má obtížnost.*

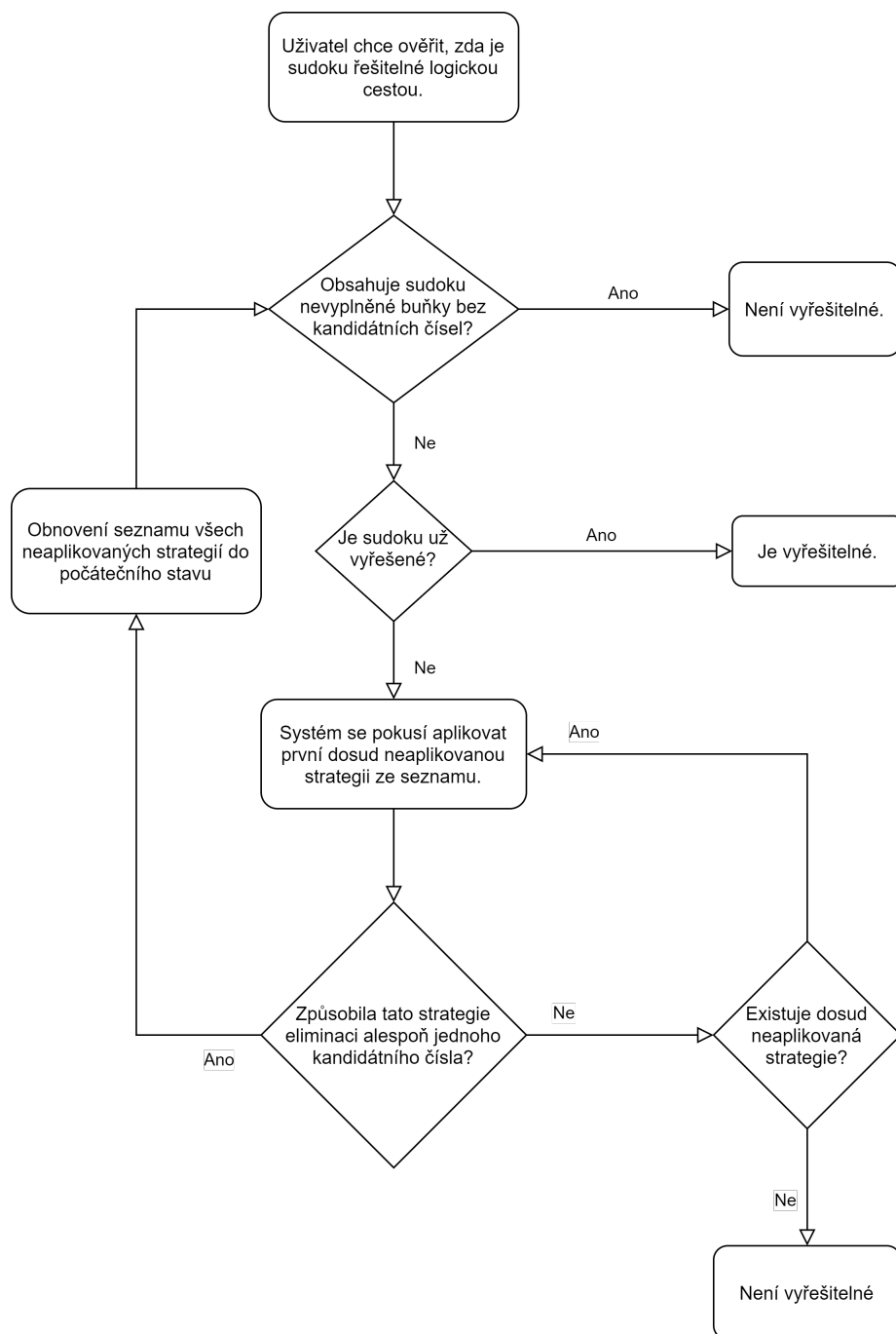
Na vývojovém diagramu 4.8 je znázorněn postup řešení ověření řešitelnosti logickou cestou. Tato metoda prochází opakovaně seznam implementovaných strategií dokud nenajde řešení, nedostane se do neřešitelné situace⁵ nebo dokud se v některé iteraci nedostane na konec seznamu strategií bez provedení změny v sudoku.

⁴Pokud nalezení dalšího kroku selže, může to znamenat, že sudoku není vůbec řešitelné, nebo že nástroj pouze neobsahuje složitější strategii, která by šla aplikovat.

⁵Indikátory neřešitelného sudoku jsou z pohledu programu dva. V prvním případě se v sudoku objeví dvě vyplněné buňky v přímé kolizi, což je situace, která může nastat pouze pokud uživatel takové sudoku odešle aplikaci. Druhou možností je, že sudoku obsahuje prázdné políčko neobsahující ani vyplněné číslo ani žádná kandidátní čísla. Prázdná buňka je chápána jako prázdná množina možných kandidátních čísel k doplnění.



Obrázek 4.7: Návrh funkce nalezení dalšího logického kroku.



Obrázek 4.8: Návrh funkce ověření řešitelnosti sudoku logickou cestou.

4.6 Kompatibilita s dalšími aplikacemi

Většina aplikací zabývajících se sudoku se omezuje pouze na určitou oblast. Tyto oblasti se často ani nepřekrývají a spolupráce mezi jednotlivými aplikacemi se vidí málokdy – ačkoliv by to pro uživatele bylo nejvýhodnější.

Zadání sudoku je přitom snadno serializovatelný soubor hodnot a jejich umístění, který by potenciálně mohl být předáván i mezi navzájem různými aplikacemi.

Tato práce řeší řešení logickou cestou a ukázání postupů, ale potřebuje k tomu zadání sudoku. Uživatel se může přijít poradit o dalším kroku nebo přijít luštit sudoku v prostředí aplikace, ale musí ručně zadat svoje zadání, než bude moct nástroj plnohodnotně využít.

Na druhou stranu existuje mnoho zdrojů se zadáními sudoku, či přímo generátory sudoku, které uživateli možnost nápovědy a řešení logickou cestou nenabízí.

Bakalářská práce studenta Juraje Lazoríka [10], která vzniká tento rok na Fakultě informačních technologií Vysokého učení technického v Brně, se zabývá naskenováním sudoku z vyfoceného zadání do mobilní aplikace, která umožní jeho doluštění. Jeho práce se zabývá sudoku z jiného úhlu pohledu než moje práce, avšak mohly by se doplnit.

Navrhli jsme tedy jednoduchý způsob propojení. Můj nástroj na řešení sudoku byl rozšířen o nový typ GET požadavku, kterému může být jako parametry předáno zadání sudoku ve formě souvislého řetězce číslic a symbolů. Jakub Lazorík do své aplikace přidal odkaz vytvořený z právě naskenovaného sudoku, který uživatele odkáže na webovou stránku a doplní do ní rozluštěné sudoku.

	1	2	3	4
1	1	2	3	
2				
3		1	2	3
4				

Obrázek 4.9: Sudoku 4x4, které se do programu nahraje při použití odkazu https://hojkas-sudoku-helper.herokuapp.com/solver/sudoku4x4?load_from=string&cells=123.....123...... Parametr *cells* obsahuje řetězec znaků délkou odpovídající počtu buněk v sudoku a popisuje jednotlivé buňky postupně po řádcích zleva doprava. Platné číslice pro dané sudoku jsou vyplněny, libovolné jiné znaky značí prázdná políčka.

Uživatel tak může využít mobilní aplikace na naskenování sudoku a logického řešení ve webové verzi bez nutnosti ručního přepisování sudoku.

Předvyplnění sudoku za pomoci parametrů odkazu může být využito více způsoby. Odkaz může být zakomponován do nástrojů zabývajících se přímo generováním sudoku, které neřeší jeho následné luštění. Zvolený formát parametru pro předání zadání sudoku v odkazu (viz obrázek 4.9) se často používá pro popsání zadání sudoku na fórech a při ukládání velkého množství zadání do textových souborů.

Jedním takovým skladištěm obrovského počtu zadání sudoku je tzv. Sudoku banka dostupná z [6]. Tento zdroj obsahuje přes 800 000 zadání klasického sudoku popsaných právě řetězcem obsahu buněk po řádcích, s číslem 0 značícím prázdnou buňku. Pro luštění těchto zadání uživatel nemusí uživatel přepisovat zadání, stačí mu zkopírovat řetězec a vložit do parametru odkazu na můj nástroj.

Kapitola 5

Implementace aplikace

Tato kapitola se zaměřuje na popis zajímavých částí implementace a problémů, kterými jsem se při implementaci zabývala. Nástroj byl vyvíjen ve frameworku Django s využitím jazyků Python a JavaScript.

5.1 Přehled implementace

Webová aplikace se dělí na dvě části – vysvětlení strategií v části „Návody“ a prostředí pro řešení sudoku s možností vyhledání dalšího logického kroku počítačem v části „Řešení sudoku“. Návody jsou statické webové stránky, zatímco řešení sudoku je interaktivní.

Pro implementaci jsem se rozhodla neuchovávat rozluštěné sudoku na straně serveru. Během luštění je sudoku kolekcí čísel vyplněných na různých místech v tabulce v prohlížeči.¹ Vyplňování sudoku mřížky uživatelem, podbarvení vybraných čísel i automatické doplnění kandidátů je implementováno v JavaScriptu na straně klienta².

Pro komunikaci se serverem během luštění je využito asynchronních požadavků pomocí ajax. Dochází k ní v následujících případech:

- *Aktualizace nastavení ovládání.* Pokud uživatel změní nastavení ovládání nástroje, je tato změna uložena do session pro zachování preferovaného nastavení i pro řešení ostatních typů sudoku.

¹Alternativním přístupem by bylo automaticky ukládat stav sudoku do session či cookies, popř. svázat stav sudoku s uživatelskými účty do databáze. Tento postup by mohl být vhodný pro nástroj, který se zaměřuje na sudoku tutoriál v podobě vypracované série lekcí, či pro aplikaci poskytující primárně generovaná sudoku a podporující uživatele v luštění pouze v ní.

Má práce se snaží soustředit více na pomoc s nalezením dalšího kroku pro sudoku, které uživatel rozluštil z jiného zdroje, a vysvětlení principů za tímto krokem. Absence uložení sudoku či uživatelské statistiky pro tento úkol není důležitá a během testování žádnému z uživatelů nechyběla.

²Při vytváření JavaScript funkcí byly převzaty následující části zdrojového kódu:

Funkce pro export tabulky z html ve formátu PNG byla převzata z [12]. Implementuje stáhnutí aktuálního stavu sudoku v sekci Řešení sudoku, sloužící uživateli pro stažení sudoku např. pro tisk či pro stažení obrázků pro návody a tuto práci, kvůli zachování konzistence grafického zvýraznění. Občasná chyba, při které se sudoku na první pokus stáhne posunuté, vychází z převzatého kódu.

Z [13] byla převzata část kódu pro vykreslování linek přes obsah tabulky. Tato funkcionalita byla využita pro grafické zvýraznění Y-Wing a XY-Chain.

Kód z [19] obsahuje kontrolu, zda je webová stránka otevřena v mobilním prohlížeči či nikoliv. Pro mobilní zařízení jsou na základě toho upravena nastavení ovládání sudoku na výchozí hodnoty vhodnější pro dotykové obrazovky.

- *Aktualizace sektorů jigsaw.* Po zadání validních sektorů jsou aktualizovány a uloženy do session. Toho je využito při vykreslení jigsaw sektorů při příštím navštívení stránky a pro správné aplikování strategií při vyhledání dalšího logického kroku.
- *Nalezení dalšího logického kroku.* Aktuální stav sudoku je transformován do JSON formátu a odeslán na server v POST požadavku. Server najde žádanou odpověď pomocí třídy `StrategyApplier` a zašle zpět JSON s informacemi o aplikované strategii, jejím principu a efektu.

Tyto informace jsou zpracovány a zobrazeny na aktuální stránku. Uživatel může krok potvrdit, čímž aplikuje změny provedené strategií, nebo lze krok zrušit. Obě možnosti vymažou informace o dané strategii z prohlížeče a vrátí grafickou podobu stránky do předchozího stavu.

- *Ověření řešitelnosti a obtížnosti sudoku.* Aktuální stav sudoku je ve formátu JSON zaslán serveru. Ten pomocí třídy `StrategyApplier` otestuje řešitelnost a obtížnost řešení logickou cestou a zašle zpátky textový popis jeho zjištění.

Opětovným voláním funkce pro nalezení dalšího kroku a jeho aplikováním lze sledovat, jak nástroj postupně řeší zadané sudoku. Tato volání nalezení dalšího kroku jsou na sobě plně nezávislá – každé pracuje s aktuálním stavem sudoku, ale „neví“, zda před ním proběhlo jiné volání a zda aktuální stav sudoku je zadaný uživatelem nebo výsledkem aplikace strategie.

Absence stavů pro postupné řešení sudoku logickými kroky programu neubírá na efektivitě. Bez ohledu na předchozí aplikovanou strategii by hledání strategie k aplikaci muselo proběhnout celé znovu – předchozí složitá strategie mohla odstraněním kandidátního čísla vytvořit podmínky pro některou snadnou strategii, kterou v předchozím hledání nešlo použít. Obdobně by zkušenost z předchozí strategie nepomohla zrychlit další vyhledávání – i kdybychom zaznamenali umístění jednotlivých charakteristických skupin hledaných strategiemi při prvním průchodu, pro druhý průchod se mohlo změnit umístění i desítek kandidátních čísel, a tyto informace nejsou nyní platné.

5.1.1 Problém hledání logických kroků

V každém validním sudoku má buňka dva možné stavy – je buď vyplněná vyřešeným číslem, nebo obsahuje množinu kandidátních čísel. Prázdná buňka se může objevit po aplikování strategie na sudoku bez řešení, kdy strategie odstraní poslední kandidátní číslo z některé buňky. Prázdná buňka je jediný způsob, jak nástroj pozná, že je sudoku neřešitelné³.

³Aplikace rozlišuje situaci neřešitelného sudoku, které nemá řešení vůbec, a pouze potenciálně neřešitelného sudoku. Na potenciálně neřešitelném sudoku selhal logický postup, protože vyčerpal seznam strategií. Takové sudoku skutečně nemusí mít řešení, nebo pouze nástroj neobsahuje dostatečně pokročilou strategii, která by vedla v pokroku.

Ve vzácných případech může jít o sudoku, které nelze vyřešit ani aplikací nejtěžších známých strategií. V takové situaci se názory komunity rozcházejí, jak je uvedeno v [17]. Nelze vyloučit, že existuje strategie, která by problém vyřešila, avšak nebyla ještě popsána. Avšak především se ke slovu dostává problém definice „řešitelnosti sudoku logickou cestou“. Např. Thomas Snyder definuje logicky řešitelné sudoku jako takové, které lze vyřešit inkoustovým perem bez mazání, s všemi odvozeními provedenými pouze v hlavě pomocí vizualizace. Další plně uznávají poznámky tužkou, avšak odsuzují strategie, které se zakládají na představení si následků určitého umístění číslice.

Se sudoku, na které nestačí běžně uváděné strategie, se běžný uživatel pravděpodobně nikdy nesetká. Debata o definici logicky vyřešitelného sudoku v extrémních případech je mimo rozsah této práce.

Pokud však sudoku zadá uživatel a nedoplní do něj kandidátní čísla, nástroj nemá prostředky pro rozpoznání, že je tato situace pouze omyl či nedorozumění. Nemá jak rozpoznat, zda tento stav nevznikl odstraněním možných kandidátních čísel. Kdyby do takového sudoku nástroj doplnil kandidátní čísla automaticky při detekci prázdného políčka, několik za sebou aplikovaných vyhledání dalšího kroku by kandidáty z něj postupně opět vyloučilo – a další zavolání by detekovalo prázdnou buňku, doplnilo by kandidáty a smýčka by pokračovala.

Proto, ačkoliv to pro uživatele znamená akci navíc, musí být před aplikací dalšího kroku nebo ohodnocením obtížnosti do sudoku doplněny kandidátní čísla. Nástroj na to obsahuje tlačítko pro usnadnění. Protože je na tento problém snadné zapomenout, při nalezení prázdného políčka při hledání dalšího kroku nástroj uživateli oznámí, že tato situace mohla vzniknout nevyřešitelností sudoku nebo opomenutím doplnění kandidátů.

5.2 Generování sudoku

Ačkoliv generování zadání sudoku není vyžadováno zadáním práce, v prvotních návrzích jsem ho plánovala zahrnout do aplikace pro naplnění většího rozsahu uživatelských potřeb očekávaných od nástroje zabývajícího se sudoku.

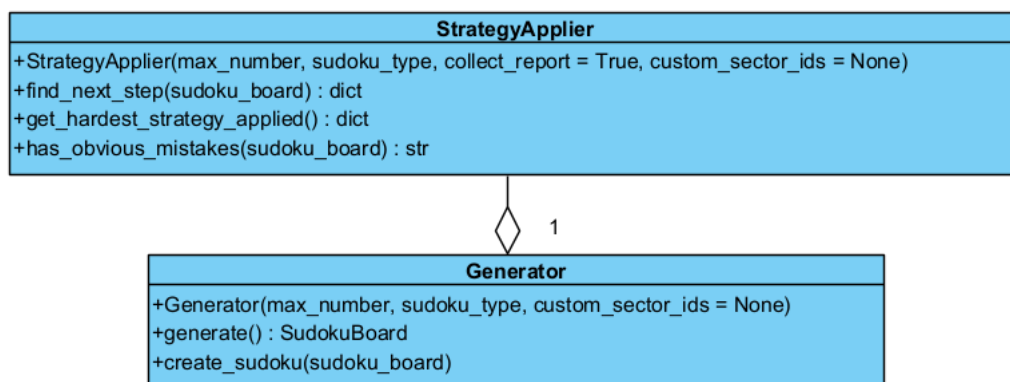
Primitivní neoptimalizovaná verze generování sudoku odhalila problém, se kterým jsem při plánování nepočítala.

Vygenerování vyplněné mřížky klasického sudoku velikosti 9x9 trvalo prototypu generátoru průměrně 1,4s. Při přidání pouhých dvou omezení na uspořádání diagonál se čas generování dostal do rozmezí 110–150s. Pro diagonální sudoku se středy čtverců (3 přidaná omezení na umístění číslic), hypersudoku (4 přidaná omezení), a sudoku 16x16 by časová náročnost byla výrazně horší.

Aby generování sudoku mohlo být plně implementováno pro všechny typy sudoku, musel by kód být rozsáhle optimalizován, implementován časově optimálnější algoritmus a kód nejlépe přepsán do jazyka nižší úrovně. Ačkoliv to není nepřekonatelná překážka, byla by taková akce časově náročná.

Protože generátor sudoku není součástí zadání práce ani primárních cílů kladených na nástroj, rozhodla jsem se tuto oblast ponechat pouze částečně implementovanou a věnovat čas zdokonalení jiných částí aplikace. Zdokonalení generátoru a jeho integrace do nástroje může být součástí budoucích vylepšení.

Prototyp generátoru sudoku jsem ponechala ve zdrojových kódech jako třídu **Generator** (diagram třídy na obrázku 5.1), příklad spuštění z Python konzole je ukázán na obrázku 5.2. Tento generátor vytváří středově symetrické sudoku, které je vyřešitelné logickou cestou za použití strategií popsaných v této práci. Pro toto ověření řešitelnosti využívá třídu **StrategyApplier**.



Obrázek 5.1: Diagram tříd pro třídy Generator a StrategyApplier. Pro přehlednost jsou uvedeny pouze parametry a metody, které slouží jako rozhraní.

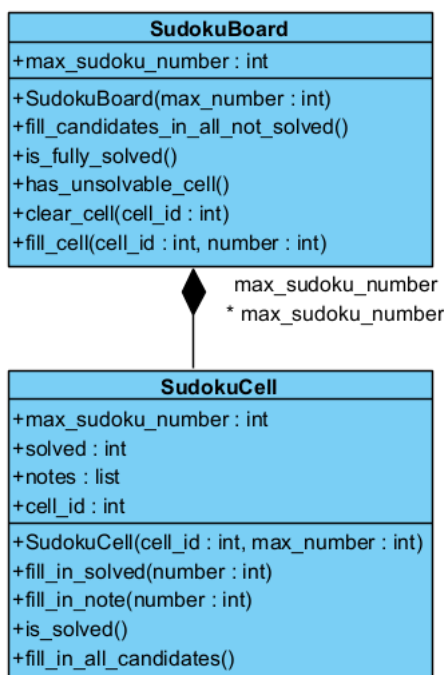
```

>>>import utils.generator
>>>gen =utils.generator.Generator(9,'classic')
>>>sudoku =gen.generate()
>>>gen.create_sudoku(sudoku)
>>>sudoku.print_full_sudoku()
#-----#-----#-----#
| 6 . 2 | . . . | . 1 . |
| . 5 4 | 6 7 1 | . 9 . |
| 3 1 . | . . . | . 8 . |
#-----#-----#-----#
| . . 5 | 9 4 7 | 2 3 . |
| 2 8 . | 3 6 5 | . 7 4 |
| . 4 3 | 1 8 2 | 9 . . |
#-----#-----#-----#
| . 7 . | . . . | . 4 3 |
| . 3 . | 5 1 6 | 7 2 . |
| . 2 . | . . . | 8 . 1 |
#-----#-----#-----#
  
```

Obrázek 5.2: Spuštění generátoru z python konzole pro klasické sudoku o velikosti 9x9 a grafická reprezentace výsledku.

5.3 Třídy popisující sudoku

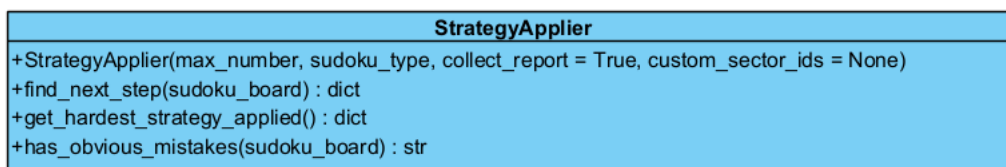
Sudoku je na straně serveru uloženo pomocí tříd znázorněných na obrázku 5.3. Protože nástroj pracuje s typy sudoku, které mají různé velikosti, mají obě třídy tuto informaci uloženou. Samotný typ pravidel sudoku zde nehraje roli.



Obrázek 5.3: Diagram tříd pro popis stavu sudoku.

5.4 Třída aplikující strategie

Třída `StrategyApplier`, jejíž rozhraní je zobrazeno v 5.4, je jádrem procesu hledání logického kroku.



Obrázek 5.4: Diagram tříd popisující třídu pro vyhledání a aplikaci strategií. Uvedeny jsou pouze parametry a metody, které slouží jako rozhraní.

Ve svém konstruktoru podle zadaného typu sudoku a jeho velikosti vytvoří mnoho pomocných proměnných pro procházení sudoku po jednotlivých blocích a pro rychlejší vyhledávání souvislostí mezi jednotlivými buňkami.

Buňky jsou v sudoku indexovány po řádcích zleva doprava pomocí id, jehož hodnoty jsou v rozmezí od 0 do velikosti sudoku na druhou (bez tohoto čísla; pro sudoku o rozměrech 9x9 by se jednalo o indexaci 0–80). Strategie sudoku ale využívají bloků – sloupců, řádků, sektorů, a dalších speciálních sektorů jako jsou diagonály, středy čtverců a sektory hypersudoku.

V konstruktoru `StrategyApplier` se proto vytvoří seznamy reprezentující jednotlivé bloky, které obsahují seznam id buněk v nich ležících⁴. Pro sudoku jigsaw je třeba, aby seznam id tvořících jednotlivé sektory byl předán jako parametr, u ostatních typů jsou umístění sektorů vypočítány. V neposlední řadě je vytvořeno také mapování jednotlivých buněk na čísla bloků, ve kterých se nachází. Toho je využito například pro odstranění kolizí kolem vyplněné buňky pro rychlejší vyhledání, které bloky buňka afektuje.

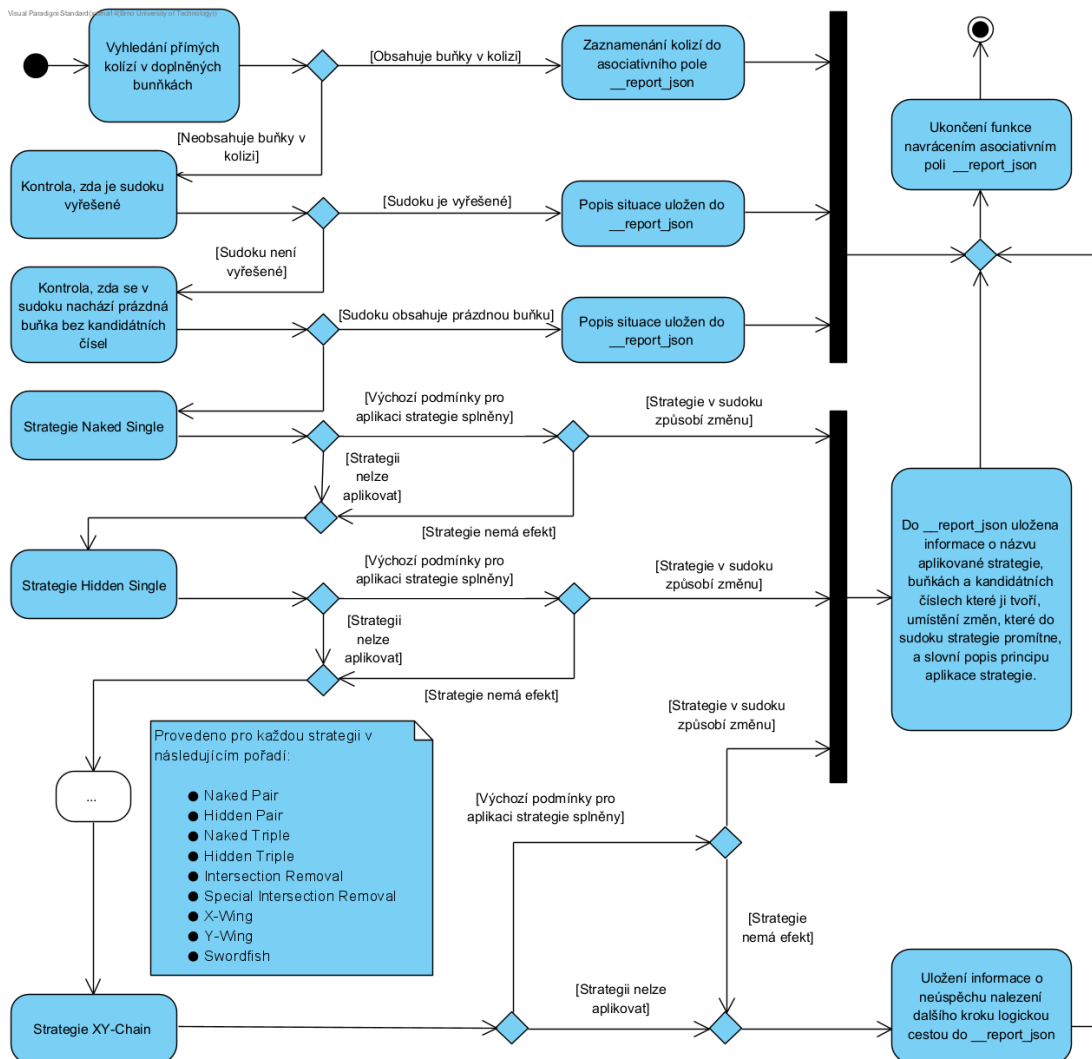
Uložení jednotlivých bloků jako seznamu id velmi zjednoduší aplikaci strategií na alternativní typy sudoku. Sudoku netradičních velikostí se od klasického liší pouze počtem identifikátorů buněk v jednotlivých blocích, což pro návrh strategie procházející seznam postupně jednu hodnotu po druhé nepředstavuje žádný rozdíl. Jigsaw má oproti klasickému sudoku 9x9 pouze jinak rozmístěné sektory (odlišné uspořádané seznamy id buněk v sektorech), což umožňuje strategiím nakládat s těmito druhy sudoku stejným způsobem. Pro zbylé druhy je rozdíl vyřešen přidáním dalších seznamů id. To umožní strategii neohlížet se na typ sudoku – pouze aplikuje svůj postup také na tyto další bloky. Ačkoliv nástroj pracuje s mnoha typy sudoku, díky těmto úpravám implementace strategií nemusí ve většině případů dělat mezi postupy pro různé typy sudoku rozdíl⁵.

Metoda `find_next_step` se stará o celý postup nalezení dalšího kroku na parametru typu `SudokuBoard`, jak je znázorněno na obrázku 5.5. Využívá k tomu řadu pomocných metod. Její výstup závisí na atributu třídy `collect_report`. Tato hodnota je předána při inicializaci třídy a slouží pro určení účelu aplikace strategií. Pokud je `true`, nalezená strategie není aplikována na předané sudoku, ale pouze zaznamená informace o aplikované strategii (či při neúspěchu) do asociativního pole a toto pole předá zpět jako návratovou hodnotu. Pokud by hodnota parametru byla `false`, nalezené strategie se přímo aplikují na předané sudoku a vrací jen boolean hodnotu reprezentující úspěch/neúspěch.

V nástroji je verze nalezení dalšího kroku se sbíráním informací použito pokud si uživatel přeje tento krok najít. Přímá aplikace logického kroku slouží pro ověření řešitelnosti a složitosti zadaného sudoku.

⁴U některých strategií nezáleží na tom, na jakém typu bloku je aplikovaná. Některé jsou však omezené na určité typy bloků (např. X-Wing zkoumá pouze vztah sloupců a řádků) a téměř všechny potřebují znát lokaci bloku kvůli vysvětlení postupu uživateli. Proto jsou seznamy id tvořících bloky uloženy samostatně pro každý typ bloku.

⁵Rozsáhlejší úpravy bylo nutné udělat pouze u Intersection Removal (nový způsob použití u diagonálního sudoku a jigsaw), u sudoku 4x4 (strategie obtížnější než Naked Pair nemá smysl hledat) a u sudoku 16x16 (kvůli své velikosti se vyplatí hledat Naked 2–8 a Hidden 2–8).



Obrázek 5.5: Diagram znázorňuje postup funkce `find_next_step` pro nalezení dalšího logického kroku, má-li třída za úkol sbírat zprávu o provedených strategiích. Element s výpustkou zastupuje provedení stejných akcí jako u strategií `Naked Single` a `Hidden Single` pro všechny strategie ze seznamu.

5.4.1 Implementace strategií

Každá strategie z kapitoly 3 je implementována jako metoda třídy `StrategyApplier` a princip každé této metody je pro každý typ sudoku totožný. Jedinou výjimku tvoří `Intersection Removal`, která je rozdělena do dvou samostatných metod kvůli speciálnímu případu použití pro jigsaw a diagonální sektory⁶.

⁶`Intersection Removal`, blíže popsany v sekci 3.6, zkoumá vztah průsečíku sektoru a sloupce (řádku). Tento průsečík tvoří buňky společné oběma blokům a pokud je jejich výskyt omezen v jednom bloku, promítne se do druhého.

`Special Intersection Removal` je situace, kdy některá buňka sudoku sdílí řádek či sloupec s všemi výskytu kandidátního čísla na diagonále či v nepravidelném jigsaw sektoru. Zatímco ostatní strategie si pro alternativní typy sudoku zachovávají stejný princip vyhledávání výchozích podmínek, `Intersection Removal` vyžaduje pro tento speciální případ rozdílný přístup k vyhledávání jeho výskytu.

Transformace každé strategie (s výjimkou těch nejjednodušších⁷) do algoritmu probíhala v následujících 4 fázích. Pro názornost budou tyto kroky blíže popsány pro strategii Naked Triple. Pro výsledný nástroj jsem podle těchto kroků implementovala všechny strategie popsané v této práci.

1. *Posbírání pomocných informací.*

Typicky procházení bloků (všech či pouze některého typu, např. pouze řádky/sloupce pro X-Wing) sudoku a zaznamenávání informací důležitých pro strategii do kombinace listů a asociativních polí. Cílem je zjednodušit následné vyhledávání podmínek pro aplikaci strategii, především zjednodušit porovnávání relevantních vlastností buněk.

Pro Naked Triple: Následující akce nástroj provádí nad každým blokem sudoku. Prozkoumává každou buňku bloku. Do seznamu asociativních polí (jedno asociativní pole pro každou buňku) zaznamená identifikátor dané buňky, seznam kandidátních čísel na dané buňce a celkový počet kandidátů. Tento seznam poté profiltruje, aby v něm zůstaly pouze ty buňky s dvěma nebo třemi kandidátními čísly.

2. *Nalezení podmínek pro aplikace strategie*

S pomocí sesbíraných informací je provedeno hledání charakteristických skupin tvořících výchozí podmínky strategie. Jsou-li tyto podmínky nalezeny, je možné strategii použít – to avšak ještě neznamená, že jejím použitím něco získáme.

Pro Naked Triple: Strategii lze aplikovat, pokud se v daném bloku nachází ve třech buňkách pouze tři různé kandidáty. Pro každý blok se porovná každá kombinace trojice nevyřešených buněk. Pokud se v této trojici nachází totožné 3 typy kandidátů (popř. v některé pouze 2, které jsou subsetem těchto 3 kandidátů), jsou splněny výchozí podmínky pro aplikaci strategie.

3. *Prozkoumání oblastí efektu strategie*

Oblast efektu je množina buněk, ve kterých strategie svou aplikací vymaže určitá kandidátní čísla. Tvar oblasti efektu vychází z povahy strategie. Nejčastěji se jedná o buňky, které sdílí blok s buňkami významnými pro aplikaci strategie. Pokud je v oblasti efektu nalezeno kandidátní číslo, které nalezená strategie může odstranit, je prohledán zbytek bloku a strategie prohlášena za úspěšnou.

Pro Naked Triple: Oblast efektu jsou všechny buňky daného bloku kromě tří, které tvoří strategii. Z oblasti efektu je možné odstranit všechna tři kandidátní čísla, která se nachází v buňkách tvořících strategii. Pokud je možné v důsledku odstranit alespoň jedno kandidátní číslo, je strategie úspěšná.

4. *Zaznamenání informací o kroku*

Pokud se má o nalezení strategie sbírat zpráva, je v tomto kroku zaznamenáno vše potřebné pro vysvětlení strategie. Pokud ne, je nalezená změna (odstranění kandidátního čísla nebo vyplnění čísla pro Naked/Hidden Single) přímo provedena v sudoku.

Pro Naked Triple: Zaznamenají se následující informace: jméno aplikované strategie, buňky a kandidátní čísla tvořící kandidáty, informace pro barevné zvýraznění jejich pozic, kandidáty odstraněné strategií a jejich barevné zvýraznění a textový popis principu tohoto logického kroku.

⁷Odstranění přímých kolizí, Naked Single a Hidden Single jsou natolik jednoduché, že nevyžadují první fázi: posbírání pomocných informací. Naked a Hidden Single navíc přeskakují také 3. fázi.

Kapitola 6

Testování aplikace

Pro účely testování byla aplikace nasazena na [Heroku](#). Heroku je založeno na modelu cloud computingu *Platform as a Service*, nebo-li platforma jako služba. V tomto modelu jsou uživateli poskytovány služby jako databáze, middleware a vývojové nástroje, přičemž veškerá infrastruktura je poskytována a spravována Heroku. Heroku se zaměřuje na usnadnění procesu nasazení a správy aplikací, podporuje mnoho moderních frameworků pro vývoj webových aplikací včetně Django a umožňuje nasazení nové verze přes Git [8].

Webová stránka je dostupná z <https://hojkas-sudoku-helper.herokuapp.com/>.

Uživatele jsem pro testování rozdělila do tří skupin:

- Technologicky zdatní uživatelé, kteří se sudoku nikdy příliš nezabývali.

Na této skupině budu zkoumat především srozumitelnost sudoku podané touto formou. Toto testování klade otázky: Nevynechávám základní aspekty sudoku a jeho luštění na úkor uživatelů, kteří s ním nemají zkušenosti? Lze se v programu a jednotlivých funkcích zorientovat i s limitovanou znalostí sudoku? Nejsou vysvětlení strategií podána příliš složitě nebo nejsou u nich nedovysvětleny základní úvahy a odvození?

- Technologicky zdatní uživatelé, kteří ve volném čase luští sudoku.

Na této skupině budu zkoumat především, zda umístění ovládacích prvků dává smysl a zda vysvětlení sudoku odpovídají jejich pohledu na sudoku. Toto testování klade otázky: Jsou strategie sudoku umístěny v odpovídajícím pořadí? Odpovídá vysvětlení strategie myšlení průměrného řešitele sudoku¹? Dává rozmístění jednotlivých funkcí programu smysl? Benefitovali by z mého nástroje i lidé v sudoku zběhlí? Používají již pro řešení sudoku některé existující aplikace?

- Uživatelé, kteří nemají s technologiemi tolik zkušeností, a kteří luští ve volném čase sudoku.

Na této skupině budu zkoumat především to, zda jsou schopní se v aplikaci vyznat i s menší zkušeností s aplikacemi. Toto testování klade otázky: Je jasné, kde lze co najít? Je stránka a především zobrazení řešeného sudoku přehledné?

U každé ze skupin dále ještě zkoumám následující otázky: Dává rozmístění ovládacích prvků smysl, je ovládání intuitivní? Vykonal daný prvek akci, kterou dle popisu očekávali?

¹Strategie v nástroji popisují dle vlastního pochopení. Tímto testováním potřebuji mimo jiné ověřit, zda je moje podání strategií srozumitelné a zda odpovídá alespoň přibližně tomu, jak to chápou ostatní řešitelé sudoku. Především jde o to zjistit, zda popsaný pohled není příliš matematický, příliš odborný nebo naopak příliš laický či zjednodušený.

Je jasné, jak nástroj ovládat a kde lze najít potřebné informace? Používali by tento nástroj, když ví, že existuje? Popřípadě kterou jeho funkci by nejpravděpodobněji využili? Co za funkce jim v nástroji chybí, co by vylepšili?

V každé skupině jsem našla 2–4 dobrovolníky, kterým jsem poslala odkaz na webovou stránku a nejprve je požádala, aby si zkusili nabízené možnosti projít sami, aby popsali své dojmy a zkušenosti, a to především ty negativní s důrazem na části, ve kterých kde byli zmatení. Po získání této zpětné vazby jsem se jich doptala na možné příčiny, proč to nebylo dobré, a sledovala jejich reakci na mnou navrhnutá možná řešení těchto problémů. Na konci jsem s každým testovacím subjektem probrala dříve zmíněné otázky, na které jsme dosud nenarazili.

6.1 Zpětná vazba od uživatelů

Mezi testovacími skupinami technicky zdatných uživatelů byly v porozumění sudoku menší rozdíly, než jsem očekávala. Všichni tuto populární hádanku znali alespoň z doslechu a chápali pravidla její klasické verze. Z alternativních druhů sudoku často neznali více než jeden nebo dva typy ani uživatelé, kteří sudoku běžně luští. Zjistila jsem, že to závisí na zdrojích, odkud zadání sudoku čerpají – většina luštitelů se drží několika oblíbených zdrojů a pokud se v nich nevyskytují jiné typy sudoku, uživatelé netouží po jejich vyhledání jinde.

6.1.1 Ovládání

Z počátku testování většině z uživatelů připadalo, že je na stránce příliš mnoho ovládacích prvků bez jiné nápovědy než bloku textu v informačním modálním okně. Místo čtení pokynů by preferovali rady, jak postupovat krok za krokem, nejlépe doplněné demonstrací v podobě obrázků nebo videa.

Uživatelům s méně zkušenostmi s technologiemi zabralo zorientování se v ovládání více práce – tento problém však mají i s ostatními nástroji pro pomoc se sudoku a je to jeden z důvodů, proč preferují řešení na papír. Pro tuto skupinu obzvlášť by byl vhodný videonávod.

Nejčastějším problémem bylo nepochopení toho, že je třeba před vyhledáním dalšího kroku doplnit do prázdných políček kandidátní čísla. Tato akce není intuitivní, avšak je nezbytná pro správné chování programu. Řešením bylo výraznější upozornění na tuto nutnou akci.

Jakmile se uživatelé s ovládáním seznámili, přišlo jim dobře navržené a ocenili především více možností zadávání čísel do sudoku, které je o mnoho pohodlnější než většina alternativ. Zdůraznili, že ovládání samo o sobě je dobré – je jen třeba ho přiblížit uživateli „stravitelnější“ formou. Chválili návrh stránky, funkčnost a množství funkcí, jež nástroje (které znají z běžného užívání) postrádají.

6.1.2 Srozumitelnost vysvětlení postupu pro řešení sudoku

Na demonstraci strategií sudoku uživatelé neměli co vytknout. Po grafické i obsahové stránce předčila jejich očekávání. Stoupenci sudoku i ti, které nikdy příliš nenadchlo, se dokázali zorientovat v prezentaci strategií a pochopit, jak jednotlivé strategie fungují a co se v sudoku jejich aplikací změnilo.

6.2 Změny aplikace na základě zpětné vazby

Na základě výsledků testování jsem nástroj v mnoha ohledech upravila. Jednalo se především o mnoho drobných úprav textů a vysvětlení, které zlepšily porozumění strategiím či lépe objasňovaly funkci ovládacích prvků. Zde jsou popsány nejzajímavější změny.

Vysvětlení, jak funguje právě otevřený typ sudoku, se dá kdykoliv během luštění otevřít v modálním okně. Do nástroje byla přidána virtuální klávesnice pro pohodlnější zadávání nejen z mobilního zařízení, ale i jako alternativní metoda zadávání pro ostatní zařízení. Ukázky sudoku pro jednotlivé strategie jsou nyní přístupné nejen ze stránky strategie, ale i jako seznam odkazů u klasického sudoku.

Vytváření sektorů pomocí podbarvování buněk je sice dobrá volba pro průměrného uživatele, ale nepraktická pro uživatele trpící barvoslepostí. Označení buněk sektoru v editaci tvaru jigsaw byla doplněna o reprezentaci symboly.

6.3 Další možná vylepšení

Nástroj jsem vyvíjela s důrazem na rozšiřitelnost do budoucna. Tato práce se nesnažila zahrnout všechny možné aspekty sudoku, avšak implementace některých dalších funkcí by rozšířila atraktivitu a užitečnost nástroje pro uživatele.

Možná vylepšení:

- Zahrnutí složitějších strategií.
- Zahrnutí více netradičních typů sudoku.
- Přidání anglické lokalizace.
- Optimalizace generátoru sudoku pro netradiční typy sudoku a intergrace do nástroje.
- Využití Sudoku banky [6] pro nahrávání klasických sudoku k řešení.
- Video návod pro rychlejší seznámení se s ovládáním.

Kapitola 7

Závěr

Cílem práce bylo vytvořit výukový program, který uživateli představí strategie logického řešení sudoku a tyto strategie implementuje do aplikace schopné najít další logický krok při řešení zadaného sudoku. Strategie a jejich aplikaci nástroj zkoumá na klasickém sudoku i na jeho populárních alternativních typech.

Vytvořila jsem systém, který mi jako řešiteli sudoku na internetu chyběl a který již nyní využívám, když se mi nedaří přijít na další krok při luštění sudoku. Výsledná webová aplikace je dostupná z <https://hojkas-sudoku-helper.herokuapp.com/>.

Tento systém v podobě webové aplikace obsahuje popis 12 strategií logického řešení sudoku demonstrujících uvažování při řešení úloh. Umožňuje uživateli řešit 9 typů sudoku, včetně jigsaw sudoku s možností zadat vlastní rozložení sektorů. U všech těchto typů si uživatel může nechat poradit další logický krok či otestovat řešitelnost logickou cestou. Vyhledání dalšího logického kroku nástroj realizuje pomocí implementovaných strategií řešení sudoku, přičemž každou strategii jsem podle vlastního návrhu převedla z textového popisu pro do programovacího jazyka.

Systém je navržen tak, aby bylo možné pomocí odkazu nahrát vlastní sudoku do sudoku mřížky. Této vlastnosti bylo využito při spolupráci se studentem Jurajem Lazoríkem. Jeho práce se zabývá naskenováním sudoku z vyfoceného zadání do mobilní aplikace, která umožní jeho doluštění. Do této aplikace přidal odkaz, který uživateli otevře mou webovou aplikaci a nahraje do ní naskenované sudoku.

Transformace logických strategií z podoby, ve které ji vnímá člověk, do programovacího jazyka, byla zajímavá výzva. Během této práce jsem si uvědomila, kolik logických odvození člověk provádí automaticky, aniž by se nad nimi pozastavil, a kolik věcí vnímá člověk v souvislostech, které se počítačem obtížně napodobují.

V práci bych ráda do budoucna pokračovala rozšířením nástroje na univerzální pomůcku pro řešení sudoku. Již nyní aplikace plní krom výukového programu také funkci prostředí pro řešení sudoku. Do budoucna by bylo pro uživatele přínosné přidat možnost generování sudoku pro alternativní typy sudoku a přeložit nástroj do angličtiny.

Literatura

- [1] ANDERSEN, L. Chapter on The history of latin squares. Department of Mathematical Sciences, Aalborg University. 2007, R-2007-32. Research Report Series. ISSN 1399-2503. Dostupné z: <https://vbn.aau.dk/ws/portalfiles/portal/13649565/R-2007-32.pdf>.
- [2] ARXIV, E. T. from the. *Mathematicians Solve Minimum Sudoku Problem* [online]. MIT Technology Review, 2012 [cit. 2021-01-20]. Dostupné z: <https://www.technologyreview.com/2012/01/06/188520/mathematicians-solve-minimum-sudoku-problem/>.
- [3] COMMONS, W. *File:Comparison Sudoku.png* — *Wikimedia Commons, the free media repository*. 2020. [Online; accessed 20-January-2021]. Dostupné z: https://commons.wikimedia.org/w/index.php?title=File:Comparison_Sudoku.png&oldid=456380959.
- [4] COMMONS, W. *File:Killersudoku bw.svg* — *Wikimedia Commons, the free media repository*. 2020. [Online; accessed 20-January-2021]. Dostupné z: https://commons.wikimedia.org/w/index.php?title=File:Killersudoku_bw.svg&oldid=464200877.
- [5] CONTRIBUTORS, S. *Solving Technique* [online]. 2020 [cit. 2021-01-21]. Dostupné z: https://www.sudopedia.org/wiki/Solving_Technique.
- [6] GRANTM. *Sudoku Exchange „Puzzle Bank“* [online]. GitHub, 2021. Dostupné z: <https://github.com/grantm/sudoku-exchange-puzzle-bank>.
- [7] *Largest multi-sudoku puzzle* [online]. Guinness World Records, 2018 [cit. 2021-01-04]. Dostupné z: <https://www.guinnessworldrecords.com/world-records/386582-largest-multi-sudoku>.
- [8] *Heroku Dev Center: Documentation* [online]. Salesforce Deevlopers, c2021 [cit. 2021-04-28]. Dostupné z: <https://devcenter.heroku.com/categories/reference>.
- [9] INTELM, H. *How To Solve Every Sudoku Puzzle (Volume 2)*. Kindle Edition. Geostar Publishing LLC., leden 2018. 275 s. Dostupné z: <https://www.amazon.com/dp/B079BTB2S9/>.
- [10] LAZORÍK, J. *Mobilní aplikace pro naskenování hry Sudoku z novin a její dohrání*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce DYK, T.

- [11] LEE, W.-M. *Programming Sudoku*. 1. vyd. Apress, 2006. Technology in Action. ISBN 9781590596623.
- [12] LYTRAS, C. *Download table as PNG using JQuery* [StackOverflow]. 2016 [cit. 2021-03-16]. Dostupné z: <https://stackoverflow.com/questions/38425931/download-table-as-png-using-jquery/40644383#answer-40644383>.
- [13] MOTTIE. *Use JQuery to draw lines over table data* [StackOverflow]. 2015 [cit. 2021-04-23]. Dostupné z: <https://stackoverflow.com/questions/33988943/trying-to-use-jquery-to-draw-lines-over-table-data#answer-33990851>.
- [14] STUART, A. *Strategy Families* [online]. 2008 [cit. 2021-4-24]. Dostupné z: https://www.sudokuwiki.org/Strategy_Families.
- [15] *Sudoku-K: Měsíčník s nápovědami logického postupu řešení* [online]. 2021 [cit. 2021-4-24]. Dostupné z: <https://www.sudoku-k.eu/>.
- [16] TETSUYA MIYAMOTO, R. F. *The KENKEN Method - Puzzles for Beginners*. 1. vyd. World Scientific, 2018. ISBN 9789813232556.
- [17] USER121. *Are there Sudoku puzzles that can't be solved logically?* [StackExchange - Puzzling]. 2014 [cit. 2021-05-04]. Dostupné z: <https://puzzling.stackexchange.com/questions/12/are-there-published-sudoku-puzzles-that-require-guessing#answer-114>.
- [18] VLASÁK, E. *Do nitra Sudoku* [online]. 2011 [cit. 2021-01-03]. Dostupné z: <http://www.vlasak.biz/sudokupdf.pdf>.
- [19] ZAPOROZHETS, M. *Detecting a mobile browser* [StackOverflow]. 2012 [cit. 2021-04-29]. Dostupné z: <https://stackoverflow.com/questions/11381673/detecting-a-mobile-browser#answer-11381730>.

Příloha A

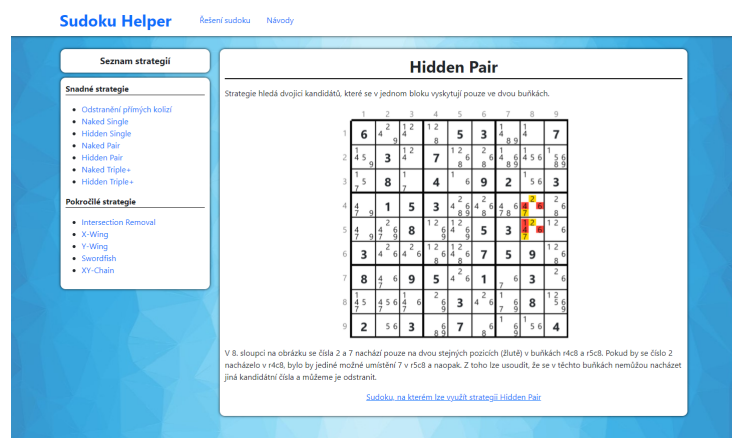
Obsah přiloženého média

Přiložené médium obsahuje následující soubory a složky:

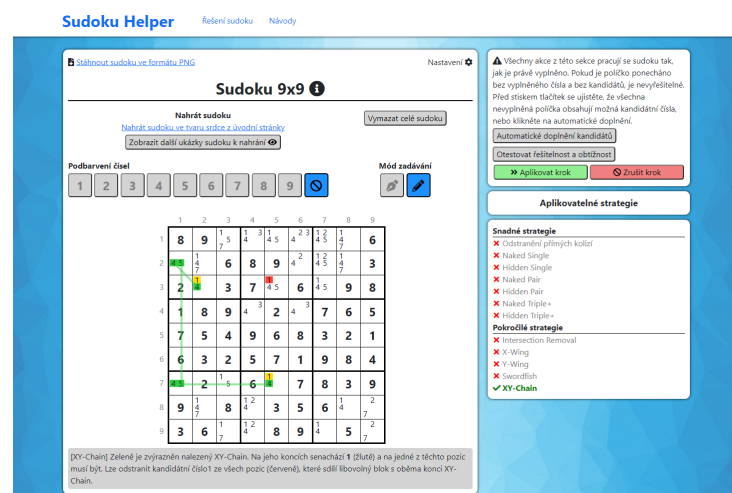
- `README.md` README soubor popisující spuštění aplikace a webovou adresu, na které je nasazena.
- `requirements.txt` Soubor popisující Python moduly, které aplikace potřebuje ke spuštění.
- `manage.py` Python soubor sloužící jako výchozí bod pro spuštění aplikace.
- `/staticfiles` Složka obsahující statické soubory: JavaScript, CSS a obrázky.
- `/sudoku` Složka se soubory definujícími chování aplikace v rámci frameworku Django.
- `/templates` Šablony pro vytvoření jednotlivých stránek v rámci webové aplikace.
- `/utils` Složka obsahující definice tříd `StrategyApplier`, `SudokuBoard`, `SudokuCell` a `Generator`.

Příloha B

Snímky obrazovky



Obrázek B.1: Stránka ze sekce „Návody“ představující uživateli strategii „Hidden Pair“.



Obrázek B.2: Stránka pro řešení klasického sudoku 9x9, kde program našel jako další logický krok strategii XY-Chain.