

SUDOKU S PŘEKRYVY: AHOJ, SVĚTE!

PAVEL STRÍŽ (CZ)

Abstrakt. Autor stručně představuje proces generování sudoku s překryvy. V pozadí používá program Sugén, který byl pro tyto účely upraven na úrovni jazyka C. Na základním příkladu je proces představen krok za krokem. V závěru autor zmiňuje omezení tohoto algoritmu. V principu můžeme říci, že jakékoliv sudoku s překryvy, které obsahuje cyklus, tímto způsobem nelze generovat, je potřeba jiný přístup, a to přes rekursivní funkci.

Klíčová slova. C, Sugén, sudoku.

MULTI-SUDOKU: HELLO, WORLD!

Abstract. The article briefly introduces a process of generating a multi-sudoku. Behind the scene, the author uses Sugén program which was modified at a C level for this specific task. A “Hello, World!” multi-sudoku is presented step-by-step. In the conclusion, the author mentions its limits, esp. which multi-sudoku cannot be generated by this algorithm. It can be stated that any multi-sudoku with cycle would be a problem, this type of multi-sudoku needs a different approach, a recursion function.

Keywords. C, Sugén, sudoku.

1. Nápad

Řeším kombinatorické úlohy, především ranking-unranking problem, tedy zjistit počet možností, vybrat si jednu z nich, zrekonstruovat kombinatorickou situaci, a naopak, z jisté kombinatorické situace spočítat její pořadové číslo. To mě přivedlo i ke klasickému sudoku, kde jsem řešení nezahlédl, ale první experimenty ukazují, že to bude náročné, ale možné. O tom snad jindy.

Při rešerši kolem sudoku jsem narazil na verze s překryvy (anglicky overlapping sudoku, multi-sudoku). Pravděpodobně nejznámější je Samurai sudoku. Ale existuje jich celá plejáda.

2. Program sugén

Chtěl jsem si něco takového zkusit naprogramovat. Při rešerši jsem narazil na program Sugén (zkráceno ze sudoku generator) od Daniela Beera, <https://dlbeer.co.nz/articles/sudoku.html>.

Ten umí vygenerovat mustry, zadání sudoku ze zadanéhoustru i získat obtížnější variantu řešeného sudoku. Neumí však generovat více sudoku s překryvy.

Poněvadž se učím C, tak jsem začal zdrojový kód zkoumat. Zkusit si vygenerovat sudoku s překryvy od nuly mě tehdy ani nenapadlo.

3. Hello, World!

Naším úkolem je připravit obdobu tohoto sudoku s překryvy. Překrývá se celý blok, to lze považovat za standard.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		4		5					2							
3		8														
4					8		7	4								
5				7	4			2		8						
6		2			7		9				6					
7		1		9			2	7								
8				1	9		4									9
9									1		8	2	7			
10			8					6	4					8	3	
11									6			2	3			
12								4				9				1
13										1	8				4	
14								9	6				4	5		
15								1	3	7			9			
16								2								6
17																

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		4	7	5	1	9	6	8	2	3						
3		8	9	6	2	4	3	5	1	7						
4		3	1	2	8	5	7	4	6	9						
5		6	3	7	4	1	5	2	9	8						
6		2	5	8	7	3	9	1	4	6						
7		1	4	9	6	8	2	7	3	5						
8		5	6	1	9	7	4	3	8	2	5	7	4	1	6	9
9		7	2	4	3	6	8	9	5	1	3	6	8	2	7	4
10		9	8	3	5	2	1	6	7	4	9	1	2	8	3	5
11								1	6	5	4	2	3	7	9	8
12								4	3	8	6	9	7	5	2	1
13								7	2	9	1	8	5	6	4	3
14								8	9	6	2	3	1	4	5	7
15								5	1	3	7	4	6	9	8	2
16								2	4	7	8	5	9	3	1	6
17																

4. Tři injekce od pana doktora (či sestřičky)

Jakmile jsem princip programu pochopil, použil jsem načtení externích souborů před zavoláním klíčové funkce. Připadal jsem si jako správný hacker.

1. injekce. Číslo sudoku slouží jako počáteční hodnota PRNG, funkce `srandom`. Pro případ pozdního nového generování jen jednoho sudoku. Nebylo třeba. Pomocný soubor obsahuje jednu hodnotu.

2. injekce. Pro vytvoření řešení, funkce `choose_grid` jsem zasahoval do pole `grid`. Pomocný soubor má dva sloupce, číslo pole (0–80) a hodnota (1–9). Vzniká soubor s řešením.

3. injekce. Pro vytvoření zadání, funkce `harden_puzzle` jsem zasahoval do pole `puzzle`. Pomocný soubor má dva sloupce, číslo pole (0–80) a jeho hodnotu (0 – musí zůstat prázdné, či 1–9). Plus se načte soubor s řešením. Výstupem je soubor se zadáním.

Poznámka. Dá se tak řešit i chtěná obtížnost, nebo program umožňuje si zavolat parametr `-t` z příkazového řádku. Toho jsem využil.

5. Dílčí kroky

Překryv je pro první sudoku 9. blok (pravý dolní roh), pro druhé sudoku blok 1. (levý horní roh). To když má člověk pořád na mysli, sledovat proces generování není pak tak náročné, hlavně u složitějších struktur.

Schématicky bychom generování mohli znázornit takto.

Upravený sugen: 1. sudoku, řešení:

-----			475196823
-----			896243517
-----			312857469
-----			637415298
-----	-->	bez injekce	-->
-----			258739146
-----			149682735
-----			561974382
-----			724368951
-----			983521674

Upravený sugen: 1. sudoku, zadání:

475196823			4_5____2_
896243517			8_____
312857469			___8_74__
637415298			__74__2_8
258739146	-->	bez injekce	-->
149682735			2__7_9__6
561974382			1_9__27__
724368951			__19_4____
983521674			_____1
			_8____6_4

Upravený sugen: 2. sudoku, řešení:

-----	382_____		382574169
-----	951_____		951368274
-----	674_____		674912835
-----	-----		165423798
-----	----	----	438697521
-----	-----		729185643
-----	-----		896231457
-----	-----		513746982
-----	-----		247859316

Upravený sugen: 2. sudoku, zadání:

382574169	000_____	_____9
951368274	001_____	__1__827__
674912835	604_____	6_4__83__
165423798	_____	_6__23__
438697521	----> _____	4__9__1
729185643	_____	__18__4__
896231457	_____	_96__45__
513746982	_____	_137__9__
247859316	_____	2_____6

Z příkazového řádku jsem si postupně dvakrát volal:

```
$ ./sugen gen-grid >reseni.txt
```

```
$ ./sugen harden -t 100 <reseni.txt >zadani.txt
```

Pro první sudoku byly dva pomocné soubory prázdné, pro druhé sudoku s naznačenými hodnotami u injekcí.

Poslední úkol je už čistě typografický, výstupní soubory si vhodně vysázet.

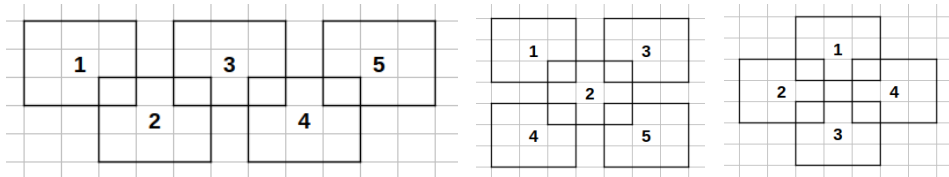
6. Kdy to nelze použít

Je zde omezení. Jakmile bychom měli komplikovanější strukturu překryvů (sudoku vztahy jakoby ob jedno sudoku), dříve či později se dostaneme do konfliktu, kdy musíme sudoku generovat znovu, či samozřejmě lépe, využít rekurze. O tom více na přednášce.

Pokud však najdeme pořadí sudoku, abychom jedno negenerovali ze vztahů více nezávislých sudoku, tato metoda nám dostačuje. Termínem z teorie grafů, do každého sudoku může jít maximálně jedna šipka, ze sudoku ven libovolný počet. Šipky určují pořadí generování.

Pro ilustraci dva náčrtky. Na levém obrázku můžeme generovat sudoku např. v těchto pořadích: 1-2-3-4-5, 5-4-3-2-1 nebo třeba 3-2-1-4-5. Do problémů se dostáváme např. při 1-2-3-5-4 či 5-1-2-3-4, ale také při pokusu generovat sudoku jakoby po řádcích, tedy 1-3-5-2-4.

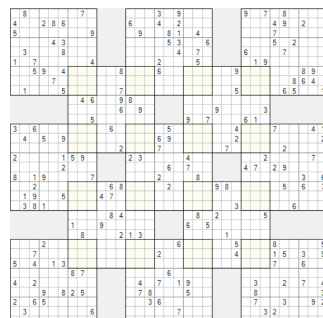
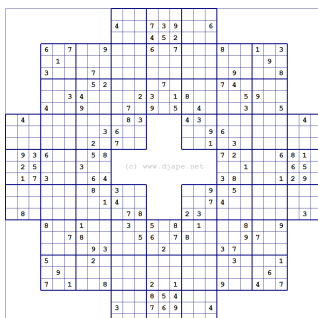
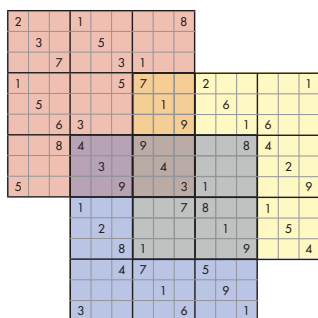
Na středním obrázku můžeme využít: 1-2-3-4-5 či 2-1-3-4-5, ale dostaneme se do problémů při 1-3-2-4-5 či 1-3-4-5-2.



Tato pomůcka sledu generování sudoku nám nepomůže u multi-sudoku, které tvoří uzavřený cyklus, kdy každé sudoku má víc nezávislých překryvů, viz pravý obrázek.

Na takovou situaci se už musí jít přes rekursivní funkci. Či mnou nedoporučený způsob vyřazování nevhodného sudoku a opakování. V takovém případě se ale může stát, že řešení ani nalézt nelze a musí se jít v procesu generování o celé sudoku nazpět. Chce to jiný přístup, o tom víc na konferenci.

Tento typ jsem neprogramoval, neb jsem to zatím nepotřeboval, ale kdyby na to došlo, zde je tzv. Venn sudoku (*Taking Sudoku Seriously*, str. 122), které je ideálním typem na testy. Plus další dva rozsáhlejší cyklické typy přebírané z internetu z různých zdrojů (Sudoku Gattai a Sudoku Sumo).



7. Rešerše zdrojů: Logická hra sudoku

Články

Rád bych upozornil na tři články:

- Arnab Kumar Maji et al.: An Exhaustive Study on Different Sudoku Solving Techniques. *International Journal of Computer Science Issues*, Vol.11, Issue 2, No.1, March 2014. ISSN 1694-0814, eISSN 1694-0784. <https://www.ijcsi.org/papers/IJCSI-11-2-1-247-253.pdf>
- Mária Ercsey-Ravasz, Zoltán Toroczkai: *The Chaos Within Sudoku*, arXiv: 1208.0370v1, August 1, 2012. <https://arxiv.org/pdf/1208.0370.pdf>
- Gary McGuire, Bastian Tugemann, Gilles Civario: *There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration*, arXiv 1201.0749v2, August 31, 2013. <https://arxiv.org/pdf/1201.0749.pdf>

Knihy

Za pozornost stojí tyto čtyři knihy.

- *Taking Sudoku Seriously*, a,
- *Geometric Magic Squares*.
- Wei-Meng Lee: *Programing Sudoku*, Apress, USA, 2006. ISBN 978-1-59059-662-3. Zdrojové kódy jsou na <https://github.com/Apress/programming-sudoku>.

- Giulio Zambon: *Sudoku Programming with C*, Apress, USA, 2015. ISBN 978-1-4842-0996-7. Zdrojové kódy jsou dostupné na <https://github.com/apress/sudoku-programming-w-c>.

Efektivní algoritmus

Efektivní algoritmus vyvinul D.E. Knuth známý jako DLX, Dancing Links, Algorithm X. Knuth, Donald E. (2000). Dancing links. *Millennial Perspectives in Computer Science*. P159. 187. arXiv 0011047v1, November 15, 2000. <https://arxiv.org/pdf/cs/0011047.pdf>

Volně dostupné programy

V Linuxu jsou k dispozici balíky:

- `sgt-puzzles`, příkaz `sgt-solo`.
- `qqwing`, příkaz ten stejný.
Umí vypsát kroky u řešení: `qqwing --generate 1 --instructions`.
- `gnome-sudoku`, příkaz stejný.
- `nbsdgames`, příkaz `nbsudoku`. Tip: `nbsudoku -s 7`.
- `sudoku`, příkaz stejný.
- `ksudoku`, příkaz stejný. Umí řadu typů, včetně 3D.
- `fltk1.3-games`, příkaz `flsudoku`.

Program Sugen byl zmíněn v tomto článku, vlastní program viz <https://dlbeer.co.nz/articles/sudoku.html>.

Na GitHubu má uživatel KyleGough svůj program `sudoku`, který řeší sudoku pomocí logických, resp. hráčských metod, viz <https://github.com/KyleGough/sudoku>.

Další zajímavé zdroje

Počty různých sudoku, viz <http://www.afjarvis.org.uk/sudoku/>.

Řešení různých variant sudoku, viz kanál na YouTube <https://www.youtube.com/@CrackingTheCryptic>.

Guinnessova kniha rekordů, <https://www.guinnessworldrecords.com/>. Zajímavé a aktivní jsou tři: Largest multi-sudoku puzzle (280 sudoku), Most people playing sudoku simultaneously (3824) a Largest sudoku published (jedná se o sudoku 100 na 100).

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,

E-mailová adresa: pavel@striz.cz