12. ročník Žilina 2. – 4. 7. 2024
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
ossconf.soit.sk
OSS Conf

online OSSConf 2024: 1–4

ÚVOD DO PROGRAMU PICAT

PAVEL STŘÍŽ (CZ)

Abstrakt. Článek stručně představuje proces ověření sudoku s překryvy, kde je nutné mít jen jedno řešení, v programu Picat. Jako ukázka slouží pět sudoku složených do podoby olympijských kruhů s překryvy velikosti jednoho bloku.

Klíčová slova. Sudoku, Picat, řešitel SAT.

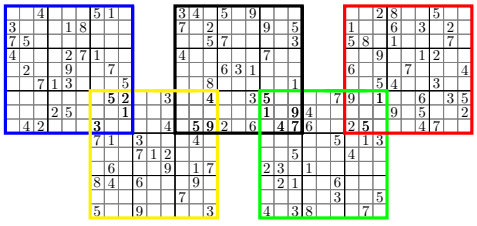
INTRODUCTION TO THE PICAT PROGRAM

Abstract. The article briefly describes a process of multi-sudoku verification, where one unique solution is necessary, in program Picat. An exemplary multi-sudoku are olympic rings formed by 5 sudokus with single block overlaps.

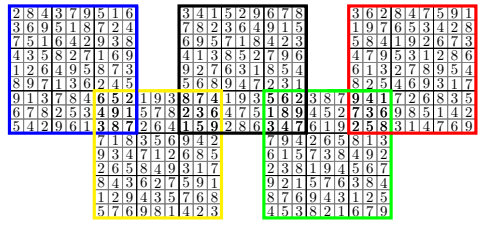
Keywords. Sudoku, Picat, SAT solver.

1. Bádání nad sudoku s překryvy

U svých experimentů nad multi-sudoku (viz příspěvek „Sudoku s překryvy: Ahoj, světe!“ v tomto sborníku) jsem se dostal do bodu, kdy jsem již byl spokojený s výstupy. A to do takové míry, že jsem připravil 5 sudoku ve tvaru olympijských kruhů a nabídl to Informačnímu bulletinu České statistické společnosti jako PF 2024 k otištění, viz IB 4/2023.



Zadání multi-sudoku.



Řešení multi-sudoku.

Ověřil jsem jedinečné řešení dílčích sudoku, přes program `sugen`, ale později také přes program `sudoku` (GitHub: KyleGough). Vynechal jsem takové, které vyžadovaly víc kol řešení, byť to asi nebylo třeba. Zkušený řešitel sudoku lítá z jednoho sudoku do druhého, vlastně takový teoretický počet nutných kol vůbec neřeší.

2. Ověření celku

Prozradím, že tento krok není potřeba, protože, když jsou dílčí sudoku s jedním řešením, musí být s jedním řešením i celek. To mě tehdy nenapadlo. Navíc šlo o mé první publikování takového ražení, chtěl jsem mít jistotu, že tam není víc řešení.

Dříve jsem zkoumal program `Picat`, kde mezi mnoha stovkami ukázek bylo i nalezení řešení klasického sudoku. Říkal jsem si, že by měl jít ověřit i mnohem větší projekt. Zajímalo mě též, jak dlouho to bude trvat.

3. Program Picat

Program spadá do kategorie SAT řešitelů, není ve standardním linuxovém repozitáři, ale dá se nainstalovat po stažení z `http://picat-lang.org`.

Syntaxe jazyka je nestandardní, ale dá se v ní rychle zorientovat. Nezvyk je, že dílčí úseky jsou oddělovány čárkami, závěr příkazu je ukončen tečkou, podobně jako složené věty v běžném jazyce.

Během generování řešení a zadání multi-sudoku se mi generuje i celý program pro Picat (užívám Lua a Python3), který se následně spustí.

4. Ukázka kódu

Generovaný soubor je ve finále obrovský, využívám `[...]` kvůli zkrácení pro účely vydání.

Soubor si pojmenujme `kruhy.pi` a část proměnných by vypadala takto. Proměnných je celkem 405, pět sudoku po 81 proměnných. Pořadí jsem měl dané zleva doprava: A pro modrou, B pro žlutou, C pro černou, D pro zelenou a E pro červenou. Zkušené oko vidí, že by se na překryvech daly proměnné ušetřit, ale kvůli čitelnosti kódu jsem to nedělal.

Tato část nám definuje proměnné, první písmeno je sudoku, první cifra řádek, druhá cifra pak sloupec daného sudoku. Proměnné omezujeme na cifry 1 až 9.

```
import sat.
main=>
Vars=[
A11,A12,A13,A14,A15,A16,A17,A18,A19,
[...]
E91,E92,E93,E94,E95,E96,E97,E98,E99
],
Vars :: 1..9,
```

Druhý obrovský logický blok je definování pravidel sudoku, tedy že se cifry 1 – 9 mohou opakovat jen jednou v řádku, sloupci a bloku jednotlivých sudoku.

```
all_different([A11,A12,A13,A14,A15,A16,A17,A18,A19]),
```

```
[...]
all_different([A11,A21,A31,A41,A51,A61,A71,A81,A91]),
[...]
all_different([A11,A12,A13,A21,A22,A23,A31,A32,A33]),
[...]
```

Další obrovský blok je definování překryvů. Tedy že jistá buňka z jednoho sudoku musí být stejná jako buňka jiného sudoku. Takto by to vypadalo pro naše olympijské kruhy.

```
A77#=B11,A78#=B12,A79#=B13,A87#=B21,A88#=B22,
  A89#=B23,A97#=B31,A98#=B32,A99#=B33,
B17#=C71,B18#=C72,B19#=C73,B27#=C81,B28#=C82,
  B29#=C83,B37#=C91,B38#=C92,B39#=C93,
C77#=D11,C78#=D12,C79#=D13,C87#=D21,C88#=D22,
  C89#=D23,C97#=D31,C98#=D32,C99#=D33,
D17#=E71,D18#=E72,D19#=E73,D27#=E81,D28#=E82,
  D29#=E83,D37#=E91,D38#=E92,D39#=E93,
```

Poslední velký blok je zapsání vlastního zadání multi-sudoku.

```
A13#=4, A17#=5, A18#=1, A21#=3, A25#=1, A26#=8,
A31#=7, A32#=5, A41#=4, A45#=2, A46#=7, A47#=1,
A52#=2, A55#=9, A58#=7, A63#=7, A64#=1, A65#=3,
A69#=5, A78#=5, A79#=2, A84#=2, A85#=5, A89#=1,
A92#=4, A93#=2, A97#=3,
[...]
```

Závěr programu obvykle tvoří výpis všech možností, nebo zjištění počtu řešení. Pro úplnost tohoto článku nechávám vypsat obojí.

```
L=findall(Vars,solve(Vars)),
writeln(L),
D=count_all(solve(Vars)),
writeln(D).
```

5. Spuštění programu

Nezbývá než vygenerovaný soubor `kruhy.pi` spustit. Z příkazového řádku by to bylo:

```
$ ./picat kruhy
```

Nebo interaktivně přímo z programu Picat:

```
$ ./picat
Picat> load("kruhy")
[...]
Picat> main
[[2,8,4,3,7,9,5,1,6,3,6,9,5,1,8,7,2,4,7,5,1,6,4,2,9,3,8,4,3,5,8,
```

```
2,7,1,6,9,1,2,6,4,9,5,8,7,3,8,9,7,1,3,6,2,4,5,9,1,3,7,8,4,6,5,2,
6,7,8,2,5,3,4,9,1,5,4,2,9,6,1,3,8,7,6,5,2,1,9,3,8,7,4,4,9,1,5,7,
8,2,3,6,3,8,7,2,6,4,1,5,9,7,1,8,3,5,6,9,4,2,9,3,4,7,1,2,6,8,5,2,
6,5,8,4,9,3,1,7,8,4,3,6,2,7,5,9,1,1,2,9,4,3,5,7,6,8,5,7,6,9,8,1,
4,2,3,3,4,1,5,2,9,6,7,8,7,8,2,3,6,4,9,1,5,6,9,5,7,1,8,4,2,3,4,1,
3,8,5,2,7,9,6,9,2,7,6,3,1,8,5,4,5,6,8,9,4,7,2,3,1,8,7,4,1,9,3,5,
6,2,2,3,6,4,7,5,1,8,9,1,5,9,2,8,6,3,4,7,5,6,2,3,8,7,9,4,1,1,8,9,
4,5,2,7,3,6,3,4,7,6,1,9,2,5,8,7,9,4,2,6,5,8,1,3,6,1,5,7,3,8,4,9,
2,2,3,8,1,9,4,5,6,7,9,2,1,5,7,6,3,8,4,8,7,6,9,4,3,1,2,5,4,5,3,8,
2,1,6,7,9,3,6,2,8,4,7,5,9,1,1,9,7,6,5,3,4,2,8,5,8,4,1,9,2,6,7,3,
4,7,9,5,3,1,2,8,6,6,1,3,2,7,8,9,5,4,8,2,5,4,6,9,3,1,7,9,4,1,7,2,
6,8,3,5,7,3,6,9,8,5,1,4,2,2,5,8,3,1,4,7,6,9]]
1
Picat> exit
```

Rychlost byla neuvěřitelná, u této úlohy hluboce pod jednu vteřinu na, řekněme, ne úplně rychlém notebooku (staříček ThinkPad T400).

6. Pár slov závěrem

Závěr naší úlohy by byl už jen čistě typografický, vysázet oněch 405 proměnných do formy olympijských kruhů. Nechávám případným badatelům na zkoumání. Důležité je, že existuje řešení, a to právě jedno, což jsme potřebovali dokázat a ověřit si.

Kontaktní adresa

Ing. Pavel Stříž, Ph.D., U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: pavel@striz.cz