Ramsey Quantifiers in Linear Arithmetics

Artifact Documentation

1 List of claims

The claims associated with Table 1 of the paper are obtained by running the following benchmarks in the file "elimination_benchmarks.py".

1. The first row in Table 1 corresponds to the function "`half_int`" in case of $\mathbb{Z}$ and "`half_real`" in case of $\mathbb{R}$ and is evaluated in Steps 4 and 5 of the evaluation instructions.

2. The second row in Table 1 corresponds to the function "`equal_exists_int`" in case of $\mathbb{Z}$ and "`equal_exists_real`" in case of $\mathbb{R}$ and is evaluated in Step 6 of the evaluation instructions.

3. The third row in Table 1 corresponds to the function "`equal_free_int`" in case of $\mathbb{Z}$ and "`equal_free_real`" in case of $\mathbb{R}$ and is evaluated in Step 7 of the evaluation instructions.

4. The fourth row in Table 1 corresponds to the function "`dickson_int`" in case of $\mathbb{Z}$ and "`dickson_real`" in case of $\mathbb{R}$ and is evaluated in Step 8 of the evaluation instructions.

5. The fifth row in Table 1 corresponds to the function "`program`" and is evaluated in Step 9 of the evaluation instructions.

6. The claim in line 1149 of the paper that the size of parameter t in φ_{half} does not have any notable effect on the computation corresponds to the functions "`half_int`" and "`half_real`" and is evaluated in Step 10 of the evaluation instructions.

7. The claim in line 1152 of the paper that the running time for φ_{program} is dominated by the Z3 satisfiability check corresponds to the function "`program`" and is evaluated in Step 11 of the evaluation instructions.

The claims associated with Table 2 of the paper are obtained by running the following benchmarks in the file "mondec_benchmarks.py".

1. The first row in Table 2 corresponds to the function "`imp_int`" in case of $\mathbb{Z}$ and "`imp_real`" in case of $\mathbb{R}$ and is evaluated in Steps 13 and 14 of the evaluation instructions.
2. The second row in Table 2 corresponds to the function "`diagonal_int`" in case of $\mathbb{Z}$ and "`diagonal_real`" in case of $\mathbb{R}$ and is evaluated in Step 15 of the evaluation instructions.
3. The third row in Table 2 corresponds to the function "`cubes_int`" in case of $\mathbb{Z}$ and "`cubes_real`" in case of $\mathbb{R}$ and is evaluated in Step 16 of the evaluation instructions.
4. The fourth row in Table 2 corresponds to the function "`cubes_int`" in case of $\mathbb{Z}$ and "`cubes_real`" in case of $\mathbb{R}$ and is evaluated in Step 17 of the evaluation instructions.
5. The fifth row in Table 2 corresponds to the function "`mixed`" and is evaluated in Step 18 of the evaluation instructions.
6. The claim in line 1198 of the paper that changing parameter k for φ_{imp}, $\varphi_{\mathrm{diagonal}}$, and φ_{mixed} does not have any notable effect on the running time corresponds to the functions "`imp_int`", "`imp_real`", "`diagonal_int`", "`diagonal_real`", and "`mixed`" and is evaluated in Step 19 of the evaluation instructions.
7. The claim in line 1207 of the paper that for large instances the running time is dominated by the satisfiability check corresponds to the function "`cubes_real`" and is evaluated in Step 20 of the evaluation instructions.

2 Download, installation, and sanity-testing

Download the artifact consisting of a "README.pdf" and the four files "ramsey.py", "elimination_benchmarks.py", "mondec_benchmarks.py", and "example.py" from `https://doi.org/10.5281/zenodo.8422415`. The artifact is also available at the GitHub repository `https://github.com/bergstraesser/ramsey-linear-arithmetics`.

Install Python 3 from `https://www.python.org/downloads`. Install Z3 (see `https://github.com/Z3Prover/z3`) for Python by running `pip install z3-solver` in the command line. To check whether the installation was successful, try to run `python` (or `python3` on some OS) followed by `from z3 import *` and `x = Int('x')` in the command line.

The artifact was tested on an Intel(R) Core(TM) i7-10510U CPU with 16GB of RAM running on Windows 10 with Python 3.11.3 and Z3 for Python with version 4.12.2 installed.

3 Evaluation instructions

1. Open the command line and go to the directory that contains the files “ramsey.py”, “elimination_benchmarks.py”, and “mondec_benchmarks.py”.
2. Start Python by typing `python` (or `python3` on some OS).
3. To start the evaluation of the claims in Table 1 of the paper, import “elimination_benchmarks.py” with `from elimination_benchmarks import *`.
4. To evaluate the first row of Table 1 for $\mathbb{Z}$, call the function `half_int(d,0)` for $d = 1, 10, 20, 50, 100$. The output `unsat` is expected. In the output `#variables input` and `#atoms input` correspond to columns 4 and 5 and `#variables output` and `#atoms output` correspond to columns 6 and 7 in the table. The running times in Table 1 (columns 8-12) correspond to `Time total` in the output.
5. To evaluate the first row of Table 1 for $\mathbb{R}$, call `half_real(d,t)` for $d = 1, 10, 20, 50, 100$ and $t = 0, 1$. For $t = 0$ (or in general $t \leq 0$) the expected output is `sat` and for $t = 1$ (or in general $t > 0$) the expected output is `unsat`.
6. To evaluate the second row of Table 1, call `equal_exists_int(d)` in case of $\mathbb{Z}$ and `equal_exists_real(d)` in case of $\mathbb{R}$ for $d = 1, 10, 20, 50, 100$. In both cases the expected output is `sat`.
7. To evaluate the third row of Table 1, call `equal_free_int(d)` in case of $\mathbb{Z}$ and `equal_free_real(d)` in case of $\mathbb{R}$ for $d = 1, 10, 20, 50, 100$. In both cases the expected output is `unsat`.
8. To evaluate the fourth row of Table 1, call `dickson_int(d)` in case of $\mathbb{Z}$ and `dickson_real(d)` in case of $\mathbb{R}$ for $d = 1, 10, 20, 50, 100$. For $\mathbb{Z}$ the expected output is `unsat` and for $\mathbb{R}$ it is `sat`. Note that in the table there is a typo: in the output for $\mathbb{R}$ the expected number of variables is $40d$ instead of $80d$.
9. To evaluate the fifth row of Table 1, call `program(d)` for $d = 1, 10, 20, 50, 100$ with the expected output `sat`. For $d = 50, 100$ we aborted the computation after 500 seconds.
10. To verify the claim in line 1149 of the paper that the size of parameter t in φ_{half} does not have any notable effect on the computation, call `half_int(1,t)` and `half_real(1,t)` for $t = 1, 100, 10000$. Note that the other parameters ℓ, u mentioned in the paper are not used anymore.
11. To verify the claim in line 1152 of the paper that the running time for φ_{program} is dominated by the Z3 satisfiability check, call `program(10)` and compare `Time elimination` and `Time total` in the output. `Time elimination` is expected to be much smaller than `Time total`.

12. To start the evaluation of the claims in Table 2 of the paper, import "mondec_benchmarks.py" with `from mondec_benchmarks import *`.
13. To evaluate the first row of Table 2 for $\mathbb{Z}$, call the function `imp_int(d,1)` for $d = 1, 5, 10, 20$. The output `Mondec:True` is expected. In the output `#variables input` and `#atoms input` correspond to columns 4 and 5 and `#variables output` and `#atoms output` correspond to columns 6 and 7 in the table. The running times in Table 2 (columns 8-11) correspond to `Time total` in the output.
14. To evaluate the first row of Table 2 for $\mathbb{R}$, call `imp_real(d,1)` for $d = 1, 5, 10, 20$. The expected output is `Mondec:False`.
15. To evaluate the second row of Table 2, call `diagonal_int(d,1)` in case of $\mathbb{Z}$ and `diagonal_real(d,1)` in case of $\mathbb{R}$ for $d = 2, 10, 20, 30$. For $\mathbb{Z}$ the expected output is `Mondec:True` and for $\mathbb{R}$ it is `Mondec:False`.
16. To evaluate the third row of Table 2, call `cubes_int(2,k,True)` in case of $\mathbb{Z}$ and `cubes_real(2,k,True)` in case of $\mathbb{R}$ for $k = 50, 100, 150, 250$. For $\mathbb{Z}$ the expected output is `Mondec:True` and for $\mathbb{R}$ it is `Mondec:False`. In case of $\mathbb{R}$ for $k = 100, 150, 250$ we aborted the computation after 500 seconds.
17. To evaluate the fourth row of Table 2, call `cubes_int(d,10,False)` in case of $\mathbb{Z}$ and `cubes_real(d,10,False)` in case of $\mathbb{R}$ for $d = 2, 10, 15, 20$. In both cases the expected output is `Mondec:True`. In case of $\mathbb{R}$ for $d = 10, 15, 20$ we aborted the computation after 500 seconds.
18. To evaluate the fifth row of Table 2, call `mixed(d,0,1)` for $d = 1, 2, 3, 4$. The expected output is `Mondec:True`.
19. To verify the claim in line 1198 of the paper that changing parameter k for φ_{imp}, $\varphi_{\text{diagonal}}$, and φ_{mixed} does not have any notable effect on the running time, call `imp_int(5,k)`, `imp_real(5,k)`, `diagonal_int(10,k)`, `diagonal_real(10,k)`, and `mixed(1,0,k)` for $k = 1, 100, 10000$. Note that k should be contained in $\mathbb{N}$ and not in $\mathbb{Z}$ as stated in the paper.
20. To verify the claim in line 1207 of the paper that for large instances the running time is dominated by the satisfiability check, call `cubes_real(2,50,True)` and compare the output `Time construction and elimination` with the output `Time sat check`. `Time construction and elimination` is expected to be much smaller.

4 Additional artifact description

The file "ramsey.py" contains the main functions "`eliminate_ramsey`" and "`is_mondec`". The function "`eliminate_ramsey(f,vars1,vars2,exvars)`" takes a quantifier-free formula f defined in Z3, lists of Z3 variables *vars1* and *vars2* of the same length and type, and a list of existentially quantified variables *exvars* and returns a quantifier-free Z3 formula f' and a list of Z3 variables $exvars'$ such that

$$\exists^{\mathsf{ram}} vars1, vars2 \colon \exists exvars \colon f \qquad \equiv \qquad \exists exvars' \colon f'.$$

Here f is assumed to be a formula in Linear Real Arithmetic, Linear Integer Arithmetic, or a decomposition of a Linear Integer Real Arithmetic formula. Moreover, it is assumed that f only uses the logical operators $\wedge$, $\vee$, and $\neg$, relations $<, \leq, >, \geq, =, \neq$, and modulo constraints are written as `s % e == t % e`. See Section 3 of the paper for definitions. For example, the following Python code (also in file "example.py") checks whether the formula $\exists^{\mathsf{ram}} x, y \colon \exists z \colon 2x < y \wedge x \leq z$ in Linear Integer Arithmetic is satisfiable.

```
from ramsey import *
x,y,z = Int('x'), Int('y'), Int('z')
f = And(2*x < y, x <= z)
f_elim, exvars_elim = eliminate_ramsey(f,[x],[y],[z])
s = Solver()
s.add(f_elim)
print(s.check())
```

The function "`is_mondec(f)`" takes a quantifier-free Z3 formula in the same format as above and returns `True` if f is monadically decomposable and `False` otherwise.

The file "elimination_benchmarks.py" calls the function "`eliminate_ramsey`" and "mondec_benchmarks.py" calls the function "`mondec_analysis`" of "ramsey.py" on example instances and outputs the result together with running time and size analysis.