The VLSAT-1 Benchmark Suite

Pierre Bouvier, Hubert Garavel

The VLSAT-1 Benchmark Suite

Pierre Bouvier, Hubert Garavel

informatics / *mathematics*

Inría

The VLSAT-1 Benchmark Suite

Pierre Bouvier, Hubert Garavel

Project-Team CONVECS

Technical Report n° 0510 — November 2020 — 6 pages

Abstract: This report presents VLSAT-1 (an acronym for "Very Large Boolean SATisfiability problems"), the first part of a benchmark suite to be used in scientific experiments and software competitions addressing SAT-solving issues. VLSAT-1 contains 100 benchmarks of increasing complexity, proposed in DIMACS CNF format under a permissive Creative Commons license. These benchmarks have been used by the 2020 International Competition on Model Counting.

Key-words: benchmark suite, Boolean satisfiability problem, data set, DIMACS CNF, Nested-Unit Petri Net, NUPN, Petri Net, SAT formula, SAT solving

Le jeu de tests VLSAT-1

Résumé : VLSAT-1 (acronyme anglais de "très grands problèmes de satisfaisabilité booléenne") est le premier volet d'une suite de tests destinée aux expérimentations scientifiques et aux compétitions de logiciels pour la résolution de problèmes SAT. VLSAT-1 contient 100 tests de complexité croissante, fournis en format DIMACS CNF sous une licence Creative Commons permissive. Ces tests ont été utilisés lors de l'édition 2020 de la compétition internationale sur le comptage de modèles.

1 Benchmark Description

VLSAT-1[1] is a collection of 100 SAT formulas, which are listed in Table 1. All these formulas are satisfiable and have been designed to accept a large number of models. Each formula is provided as a a a separate file, expressed in Conjunctive Normal Form and encoded in the DIMACS-CNF format[2]. Each file is then compressed using bzip2 to save disk space and allow faster downloads. The 100 formulas require 2.1 gigabytes of disk space and 419 megabytes when compressed using bzip2.

2 Scientific Context

These formulas have been generated as a by-product of our recent work [1] on the decomposition of Petri nets into networks of automata, a problem that has been around since the early 70s. Concretely, we developed a tool chain that takes as input a Petri net (which must be ordinary, safe, and hopefully not too large) and produces as output a network of automata that execute concurrently and synchronize using shared transitions. Precisely, this network is expressed as a *Nested-Unit Petri Net* (NUPN) [4], i.e., an extension of a Petri net, in which places are grouped into sets (called *units*) that denote sequential components. A NUPN provides a proper structuration of its underlying Petri net, and enables formal verification tools to be more efficient in terms of memory and CPU time. Hence, the NUPN concept has been implemented in many tools and adopted by software competitions, such the Model Checking Contest[3] [7, 6] and the Rigorous Examination of Reactive Systems challenge[4] [5, 8]. Each NUPN generated by our tool chain is *flat*, meaning that its units are not recursively nested in each other, and *unit-safe*, meaning that each unit has at most one execution token at a time.

Our tool chain works by reformulating concurrency constraints on Petri nets as logical problems, which can be later solved using third-party software, such as SAT solvers, SMT solvers, and tools for graph colouring and finding maximal cliques [1]. We applied our approach to a large collection of more than 12,000 Petri nets from multiple sources, many of which related to industrial problems, such as communication protocols, distributed systems, and hardware circuits. We thus generated a huge collection of SAT formulas, from which we carefully selected a subset of 100 formulas for VLSAT. Figure 1 shows the scalability of our benchmark suite, which properly represents the diversity of our experiments.

3 Structure of VLSAT Formulas

Each of our formulas was produced for a particular Petri net. A formula depends on three factors:

- the set P of the places of the Petri net;

- a *concurrency relation* $\parallel$ defined over P such that $p \parallel p'$ is the two places p and p' may simultaneously have an execution token; and

- a chosen number n of units.

---

[1] https://cadp.inria.fr/resources/vlsat
[2] http://www.satcompetition.org/2009/format-benchmarks2009.html
[3] https://mcc.lip6.fr
[4] http://rers-challenge.org

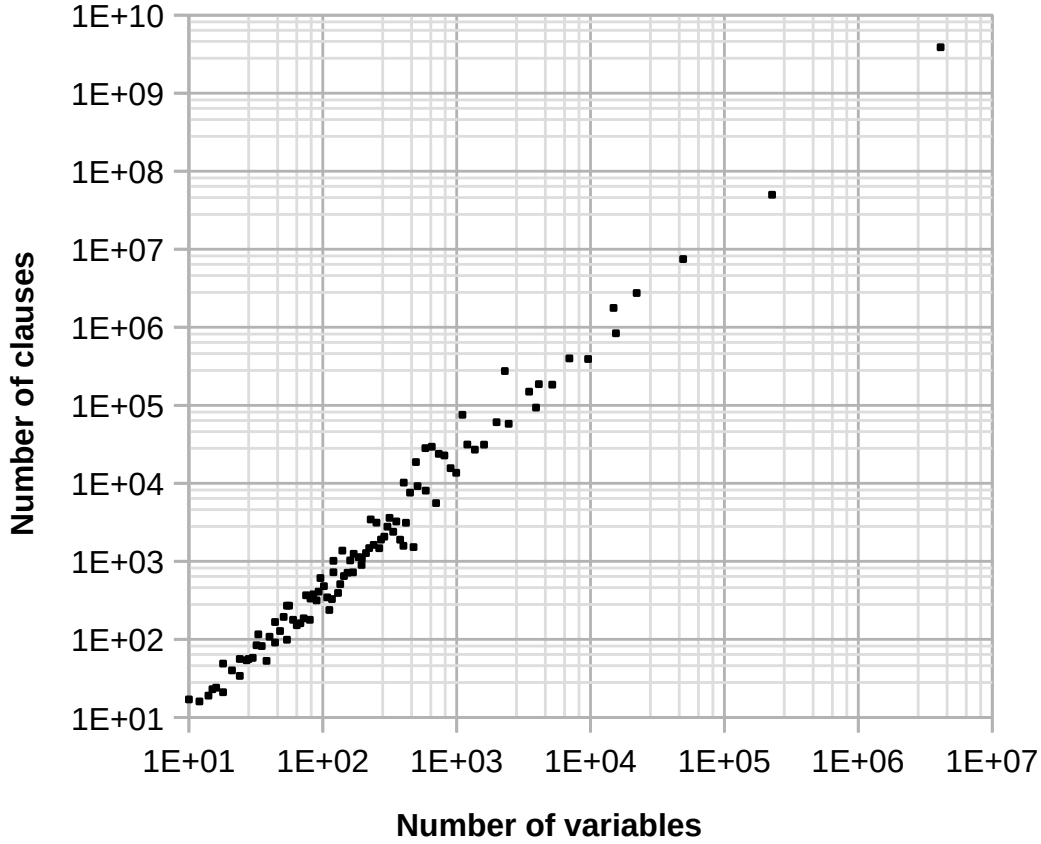| no.	variables	clauses	no.	variables	clauses	no.	variables	clauses
1	10	17	35	93	409	68	402	10,189
2	12	16	36	96	611	69	418	3119
3	14	19	37	102	481	70	448	7592
4	15	23	38	108	345	71	476	1523
5	16	24	39	112	238	72	496	18,680
6	18	21	40	117	328	73	510	9201
7	18	49	41	120	724	74	584	28,218
8	21	40	42	120	1016	75	588	8050
9	24	34	43	130	393	76	652	29,387
10	24	56	44	135	510	77	702	5565
11	27	54	45	140	1371	78	735	23,842
12	28	56	46	144	648	79	810	22,731
13	30	58	47	153	714	80	900	15,616
14	32	84	48	160	1028	81	992	13,641
15	33	116	49	168	722	82	1104	75,598
16	35	82	50	170	1247	83	1200	31,325
17	38	53	51	185	1132	84	1365	26,963
18	40	108	52	195	899	85	1600	31,240
19	44	91	53	196	1092	86	1984	60,716
20	44	167	54	210	1275	87	2289	274,818
21	48	128	55	222	1477	88	2450	58,066
22	51	194	56	228	3437	89	3480	149,734
23	54	99	57	240	1624	90	3920	93,576
24	54	270	58	252	3132	91	4114	186,615
25	56	270	59	264	1474	92	5184	184,104
26	60	178	60	272	1898	93	6954	399,521
27	64	152	61	288	2066	94	9588	392,364
28	68	161	62	304	2782	95	14,847	1,769,105
29	72	186	63	315	3608	96	15,498	838,393
30	75	367	64	336	2394	97	22,110	2,753,207
31	80	178	65	354	3239	98	49,200	7,490,695
32	81	333	66	378	1893	99	227,046	49,947,755
33	85	377	67	400	1580	100	4,114,810	3,879,649,625
34	90	316						

Table 1: List of VLSAT-1 formulas

Figure 1: Dispersion of the VLSAT formulas

A formula expresses whether there exists a partition of P into n subsets P_i ($1 \leq i \leq n$) such that, for each i, and for any two places p and p' of P_i, $p \neq p' \implies \neg (p \,\|\, p')$. A model of this formula is thus an allocation of places into n units, i.e., a valid decomposition of the Petri net. The value of n is chosen large enough so that the formula is satisfiable, i.e., at least one decomposition exists. This can also be seen as an instance of the graph coloring problem, in which n colors are to be used for the graph with vertices defined by the places of P and edges defined by the concurrency relation.

More precisely, each formula was generated as follows. For each place p and each unit u, we created a propositional variable x_{pu} that is true iff place p belongs to unit u. We then added constraints over these variables:

- For each unit u and each two places p and p' such that $p \,\|\, p'$ and $\#p < \#p'$, where $\#p$ is a bijection from places names to the interval $[1, \mathrm{card}\,(P)]$, we added the constraint $\neg x_{pu} \lor \neg x_{p'u}$ to express that two concurrent places cannot be in the same unit.

- For each place p, we could have added the constraint $\bigvee_u x_{pu}$ to express that p belongs to at least one unit, but this constraint was too loose and allowed $n!$ similar solutions, just by permuting unit names. We thus replaced this constraint by a stricter one that

breaks the symmetry between units: for each place p, we added the refined constraint $\bigvee_{1 \leq \#u \leq min(\#p,n)} x_{pu}$, where $\#u$ is a bijection from unit names to the interval $[1, n]$.

Application

The VLSAT-1 benchmarks are licensed under the CC-BY Creative Commons Attribution 4.0 International License[5].

Some of the VLSAT-1 formulas have been reused (after some scrambling) in the Instances Track 1 of the Model Counting 2020 Competition [2, 3].

Acknowlegements

References

[1] Pierre Bouvier, Hubert Garavel, and Hernán Ponce de León. Automatic Decomposition of Petri Nets into Automata Networks – A Synthetic Account. In Ryszard Janicki, Natalia Sidorova, and Thomas Chatain, editors, *Proceedings of the 41st International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS'20), Paris, France*, volume 12152 of *Lecture Notes in Computer Science*, pages 3–23. Springer, June 2020.

[2] Johannes K. Fichte and Markus Hecher. MCC 2020 - Model Counting Competition. https://mccompetition.org/assets/files/2020/MC2020_awards.pdf, July 2020.

[3] Johannes K. Fichte, Markus Hecher, and Florim Hamiti. The Model Counting Competition 2020. Technical Report, November 2020.

[4] Hubert Garavel. Nested-Unit Petri Nets. *Journal of Logical and Algebraic Methods in Programming*, 104:60–85, April 2019.

[5] Marc Jasper, Maximilian Fecke, Bernhard Steffen, Markus Schordan, Jeroen Meijer, Jaco van de Pol, Falk Howar, and Stephen F. Siegel. The RERS 2017 Challenge and Workshop. In Hakan Erdogmus and Klaus Havelund, editors, *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software (SPIN'17), Santa Barbara, CA, USA*, pages 11–20. ACM, July 2017.

[6] Fabrice Kordon, Hubert Garavel, Lom Messan Hillah, Emmanuel Paviot-Adet, Loïg Jezequel, Francis Hulin-Hubard, Elvio Amparore, Marco Beccuti, Bernard Berthomieu, Hugues Evrard, Peter G. Jensen, Didier Le Botlan, Torsten Liebke, Jeroen Meijer, Jiří Srba, Yann Thierry-Mieg, Jaco van de Pol, and Karsten Wolf. MCC'2017 – The Seventh Model Checking Contest. *Transactions on Petri Nets and Other Models of Concurrency*, XIII:181–209, 2018.

[7] Fabrice Kordon, Hubert Garavel, Lom Messan Hillah, Emmanuel Paviot-Adet, Loïg Jezequel, César Rodríguez, and Francis Hulin-Hubard. MCC'2015 – The Fifth Model Checking Contest. *Transactions on Petri Nets and Other Models of Concurrency*, XI:262–273, 2016.

[8] Bernhard Steffen, Marc Jasper, Jeroen Meijer, and Jaco van de Pol. Property-Preserving Generation of Tailored Benchmark Petri Nets. In *Proceedings of the 17th International Conference on Application of Concurrency to System Design (ACSD'17), Zaragoza, Spain*, pages 1–8. IEEE Computer Society, June 2017.

---

[5]License terms available from http://creativecommons.org/licenses/by/4.0
[6]https://www.grid5000.fr