SAT-Competition Benchmarks Spawning from Concurrency Theory

Pierre Bouvier and Hubert Garavel
Univ. Grenoble Alpes, INRIA, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{pierre.bouvier,hubert.garavel}@inria.fr

***Abstract*—We present an original approach for generating Boolean formulas stemming from the decomposition of Petri nets into automata networks. Carefully chosen examples of these formulas have been proposed for the 2020 and 2021 editions of the SAT Competition.**

I. Scientific Context

Interesting Boolean formulas can be 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) [2], 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 structuring 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[1] [3], [4] and the Rigorous Examination of Reactive Systems challenge[2] [5], [6], [7]. 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 coloring 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 Boolean formulas, from which we carefully selected a subset of formulas matching the requirements of the SAT Competition.

II. Structure of 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.

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, $p \neq p' \implies \neg\,(p \parallel p')$. 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 $p \parallel p'$ and $\#p < \#p'$, where $\#p$ is a bijection from places names to the interval $[1, \mathrm{card}\,(P)]$, we added the constraint $\neg x_{pu} \vee \neg x_{p'u}$ 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 $[1, n]$.

Each formula is provided as a separate file, expressed in Conjunctive Normal Form and encoded in the DIMACS-CNF format[3].

III. Selection of Benchmarks

Using the approach presented in Sections I and II, we previously published a test suite, named VLSAT1 [8], of 100 formulas. However, VLSAT1 only contains satisfiable formulas, as it was designed for the Model Counting Competition, which seeks formulas accepting a large number of models.

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

For the SAT Competition, we therefore undertook the production of a different collection containing both satisfiable and unsatisfiable formulas, depending on the number of units chosen for a given Petri net. Figure 1 shows that, despite the symmetry-breaking constraints mentioned in Sect. II, satisfiable formulas are often easier to solve than unsatisfiable ones.

For the SAT 2020 Competition, we submitted 36 formulas, listed in Table I. All of them have been tagged as "interesting" by the organizers of the competition, who selected 7 satisfiable and 7 unsatisfiable formulas for the Main Track; the selected formulas are those marked with a star in this table.

TABLE I
LIST OF 2020 FORMULAS

type	*variables*	*clauses*	*type*	*variables*	*clauses*
SAT*	16,676	1,598,591	UNSAT	14,640	1,323,246
SAT	18,090	1,781,277	UNSAT*	15,440	1,409,906
SAT	20,868	2,204,462	UNSAT*	15,960	1,464,039
SAT	21,190	2,597,791	UNSAT	16,297	1,562,268
SAT	21,573	2,289,124	UNSAT	17,688	1,741,702
SAT*	24,450	2,770,239	UNSAT	20,424	2,157,568
SAT	26,606	3,191,844	UNSAT*	21,114	2,240,429
SAT	27,507	3,314,450	UNSAT	23,961	2,714,844
SAT	29,736	3,780,419	UNSAT	26,104	3,131,630
SAT*	30,744	3,925,645	UNSAT	26,988	3,251,923
SAT	33,040	4,437,242	UNSAT	29,205	3,712,921
SAT*	34,161	4,607,712	UNSAT*	30,195	3,855,554
SAT	36,518	5,166,057	UNSAT*	32,480	4,362,044
SAT*	37,758	5,364,539	UNSAT	33,582	4,529,625
SAT*	40,170	5,970,608	UNSAT*	35,929	5,082,743
SAT	41,535	6,200,014	UNSAT*	39,552	5,878,762
SAT*	57,038	10,572,502	UNSAT	40,896	6,104,639
SAT	71,816	14,478,832			
SAT	83,334	20,350,783			

For the SAT 2021 Competition, we submit 20 formulas, 10 satisfiable and 10 unsatisfiable ones, which are listed in Table II. All of them have been checked by five solvers (CaDiCal, MathSAT, MiniSAT, Kissat and Z3) in their most recent versions. We used a machine with a Xeon E5-2630 v3 and 128 GB RAM. Each satisfiable formula takes at least 35 seconds with any of these solvers. Each unsatisfiable formula takes at least 37 minutes with any of these solvers.

TABLE II
LIST OF 2021 FORMULAS

type	*variables*	*clauses*	*type*	*variables*	*clauses*
SAT	11,130	1,186,888	UNSAT	1134	26,703
SAT	11,374	1,150,943	UNSAT	1155	42,917
SAT	19,565	3,665,001	UNSAT	4424	545,056
SAT	29,736	3,780,419	UNSAT	5152	824,642
SAT	37,758	5,364,539	UNSAT	5600	1,042,700
SAT	59,204	10,973,962	UNSAT	11,280	4,223,777
SAT	67,996	13,708,722	UNSAT	11,664	5,532,624
SAT	68,760	13,862,744	UNSAT	14,280	6,781,327
SAT	69,524	14,016,766	UNSAT	14,424	7,585,190
SAT	70,288	14,170,788	UNSAT	16,788	9,021,307

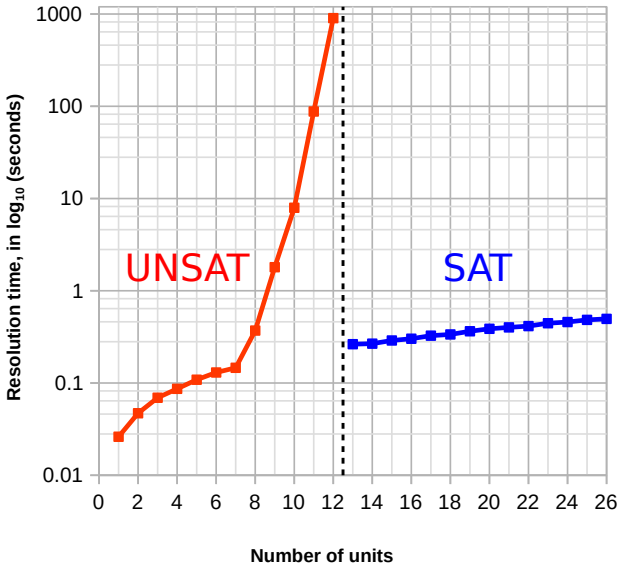

Fig. 1. Resolution times for a given NUPN

REFERENCES

[1] P. Bouvier, H. Garavel, and H. P. de León, "Automatic Decomposition of Petri Nets into Automata Networks – A Synthetic Account," in *Proceedings of the 41th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS'20), Paris, France*, ser. Lecture Notes in Computer Science, R. Janicki and N. Sidorova, Eds. Springer, Jun. 2020.

[2] H. Garavel, "Nested-Unit Petri Nets," *Journal of Logical and Algebraic Methods in Programming*, vol. 104, pp. 60–85, Apr. 2019.

[3] F. Kordon, H. Garavel, L. M. Hillah, E. Paviot-Adet, L. Jezequel, C. Rodríguez, and F. Hulin-Hubard, "MCC'2015 – The Fifth Model Checking Contest," *Transactions on Petri Nets and Other Models of Concurrency*, vol. XI, pp. 262–273, 2016.

[4] F. Kordon, H. Garavel, L. Hillah, E. Paviot-Adet, L. Jezequel, F. Hulin-Hubard, E. Amparore, M. Beccuti, B. Berthomieu, H. Evrard, P. G. Jensen, D. Le Botlan, T. Liebke, J. Meijer, J. Srba, Y. Thierry-Mieg, J. van de Pol, and K. Wolf, "MCC'2017 – The Seventh Model Checking Contest," *Transactions on Petri Nets and Other Models of Concurrency*, vol. XIII, pp. 181–209, 2018.

[5] M. Jasper, M. Fecke, B. Steffen, M. Schordan, J. Meijer, J. van de Pol, F. Howar, and S. F. Siegel, "The RERS 2017 Challenge and Workshop," in *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software (SPIN'17), Santa Barbara, CA, USA*, H. Erdogmus and K. Havelund, Eds. ACM, Jul. 2017, pp. 11–20.

[6] B. Steffen, M. Jasper, J. Meijer, and J. 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*. IEEE Computer Society, Jun. 2017, pp. 1–8.

[7] M. Jasper, M. Mues, A. Murtovi, M. Schlüter, F. Howar, B. Steffen, M. Schordan, D. Hendriks, R. R. H. Schiffelers, H. Kuppens, and F. W. Vaandrager, "RERS 2019: Combining Synthesis with Real-World Models," in *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19), Part III: TOOLympics, Prague, Czech Republic*, D. Beyer, M. Huisman, F. Kordon, and B. Steffen, Eds. Springer, Apr. 2019, pp. 101–115.

[8] P. Bouvier and H. Garavel, "The VLSAT-1 Benchmark Suite," INRIA Grenoble Rhône-Alpes, Tech. Rep., Nov. 2020.