Four Flavors of Entailment for Projected Model Counting

Sibylle Möhle[1], Roberto Sebastiani[2], and Armin Biere[1]

[1] Johannes Kepler University Linz, Austria
{sibylle.moehle-rotondi, biere}@jku.at
[2] DISI, University of Trento, Italy
roberto.sebastiani@unitn.it

Abstract

Based on our work accepted at SAT'20, we present a novel approach for enumerating partial models of a propositional formula, inspired by how the theory and SAT solvers interact in lazy SMT. Using various forms of dual reasoning allows our CDCL-based algorithm to enumerate partial models without exploring and shrinking full models. Chronological backtracking renders the use of blocking clauses obsolete. Our focus is on projected model enumeration without repetition, hence adapting it to support projected model counting is straightforward. In this presentation-only talk, we introduce the key ideas with focus on the formalization.

The task of computing the number of models of a propositional formula, also referred to as #SAT, is used, e.g., in verification [3, 4, 5, 6, 7], reasoning [14, 2, 10], diagnosis [8], and planning [1, 18]. We define the model count of a formula F as the number of its *total* satisfying assignments. A *partial* satisfying assignment I, i.e., a model in which some variables remain unassigned, therefore represents a set of total models of F. We call the number of total models of F represented by I the *model count of F under I*. The model count of F equals the sum of the model counts of F under its (possibly partial) pairwise contradicting satisfying assignments.

If only a subset X of the variables is significant, then the models are *projected* onto these *relevant* variables. We say that we *existentially quantify* the formula over the *irrelevant* variables Y and write $\exists Y\, [\, F(X, Y)\,]$, where $F(X, Y)$ is a formula over the sets of variables X and Y such that $X \cap Y = \emptyset$. Projected model counting is applied in product configuration [19] and planning [1, 18]. Recently, different approaches have been presented to address exact projected model counting. Our previous approach [11] is based on dual reasoning and enables the detection of partial models. Lagniez and Marquis [9] presented a recursive approach, while Sharma et al. [16] extended the solver sharpSAT [17] with projection capabilities.

Similarly to the non-projected case, the model count of $\exists Y\, [\, F(X, Y)\,]$, i.e., the model count of F projected onto X, equals the sum of the model counts of $\exists Y\, [F(X, Y)\,]$ under its (possibly partial) pairwise contradicting satisfying assignments projected onto X. To determine it, we first compute the pairwise contradicting partial models of F projected onto X using the algorithm presented in our work [13] accepted at SAT'20. In that work, we enumerate those models without repetition. Our method is inspired by how the theory and SAT solvers interact in lazy SMT [15]. Our basic idea was to detect partial assignments entailing the formula on-the-fly. We present four entailment tests of different strength and computational cost and a formal calculus extending our previous one [12]. Consider the formula $F = (x \wedge y) \vee (x \wedge \neg y)$ over variables $X = \{x\}$ and $Y = \{y\}$, and $I = x$ ranging over $X \cup Y$. Clearly I entails F, but a SAT solver cannot detect that. Combining dual reasoning with oracle calls allows avoiding shrinking total models. Finally, by adopting chronological CDCL we circumvent the use of blocking clauses. Our algorithm [13] yields pairwise contradicting partial models. Its adaptation to support exact projected model counting is therefore straightforward and requires only slight modifications. In essence, instead of recording partial models, we directly sum up their model counts. In this presentation-only talk, we introduce the key ideas with focus on their formalization.

Acknowledgments. The work was supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria, by the QuaSI project funded by D-Wave Systems Inc., and by the Italian Assocation for Artificial Intelligence (AI*IA). We thank Mathias Fleury for suggesting textual improvements.

References

[1] Rehan Abdul Aziz, Geoffrey Chu, Christian J. Muise, and Peter J. Stuckey. #∃SAT: Projected model counting. In *SAT*, volume 9340 of *LNCS*, pages 121–137. Springer, 2015.

[2] Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Solving #SAT and Bayesian inference with backtracking search. *J. Artif. Intell. Res.*, 34:391–442, 2009.

[3] Fabrizio Biondi, Michael A. Enescu, Annelie Heuser, Axel Legay, Kuldeep S. Meel, and Jean Quilbeuf. Scalable approximation of quantitative information flow in programs. In *VMCAI*, volume 10747 of *LNCS*, pages 71–93. Springer, 2018.

[4] Jan Burchard, Dominik Erb, and Bernd Becker. Characterization of possibly detected faults by accurately computing their detection probability. In *DATE*, pages 385–390. IEEE, 2018.

[5] Linus Feiten, Matthias Sauer, Tobias Schubert, Alexander Czutro, Eberhard Böhl, Ilia Polian, and Bernd Becker. #SAT-based vulnerability analysis of security components - A case study. In *DFT*, pages 49–54. IEEE Computer Society, 2012.

[6] Linus Feiten, Matthias Sauer, Tobias Schubert, Victor Tomashevich, Ilia Polian, and Bernd Becker. Formal vulnerability analysis of security components. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(8):1358–1369, 2015.

[7] Vladimir Klebanov, Norbert Manthey, and Christian J. Muise. SAT-based analysis and quantification of information flow in programs. In *QEST*, volume 8054 of *LNCS*, pages 177–192. Springer, 2013.

[8] T.K. Satish Kumar. A model counting characterization of diagnoses. In *DX 02*, pages 70–76, 2002.

[9] Jean-Marie Lagniez and Pierre Marquis. A recursive algorithm for projected model counting. In *AAAI*, pages 1536–1543. AAAI Press, 2019.

[10] Wei Li, Peter van Beek, and Pascal Poupart. Performing incremental Bayesian inference by dynamic model counting. In *AAAI*, pages 1173–1179. AAAI Press, 2006.

[11] Sibylle Möhle and Armin Biere. Dualizing projected model counting. In *ICTAI*, pages 702–709. IEEE, 2018.

[12] Sibylle Möhle and Armin Biere. Combining conflict-driven clause learning and chronological backtracking for propositional model counting. In *GCAI*, volume 65 of *EPiC Series in Computing*, pages 113–126. EasyChair, 2019.

[13] Sibylle Möhle and Armin Biere. Four flavors of entailment. In Luca Pulina and Martina Seidl, editors, *SAT'20*, LNCS. Springer, to appear.

[14] Dan Roth. On the hardness of approximate reasoning. *Artif. Intell.*, 82(1-2):273–302, 1996.

[15] Roberto Sebastiani. Lazy Satisfiability Modulo Theories. *JSAT*, 3(3-4):141–224, 2007.

[16] Shubham Sharma, Subhajit Roy, Mate Soos, and Kuldeep S. Meel. GANAK: A scalable probabilistic exact model counter. In *IJCAI*, pages 1169–1176. ijcai.org, 2019.

[17] Marc Thurley. sharpSAT - counting models with advanced component caching and implicit BCP. In *SAT*, volume 4121 of *LNCS*, pages 424–429. Springer, 2006.

[18] Erik Peter Zawadzki, André Platzer, and Geoffrey J. Gordon. A generalization of SAT and #SAT for robust policy evaluation. In *IJCAI*, pages 2583–2590. IJCAI/AAAI, 2013.

[19] Christoph Zengler and Wolfgang Küchlin. Boolean quantifier elimination for automotive configuration - A case study. In *FMICS*, volume 8187 of *LNCS*, pages 48–62. Springer, 2013.