Speeding up SAT for EDA

Slawomir Pilarski and Gracia Hu

Synopsys, Inc. 19500 NW Gibbs Dr, Beaverton OR 97006

ABSTRACT

This paper presents performance results for a new SAT solver designed specifically for EDA applications. The new solver significantly outperforms most efficient SAT solvers – Chaff[2], SATO[3], and GRASP[1] – on a large set of benchmarks. Performance improvements for standard benchmark groups vary from 1.5x to 60x. They were achieved through a new decision-making strategy and more efficient boolean constraint propagation (*BCP*).

1. BCP. In a typical circuit-derived CNF formula, 55%-85% of all clauses have two literals. Note that a 0-assignment to a watched literal in a two-literal clause is an *immediate implication*, and there is no need to search the clause for non-watched literals, as it happens in Chaff or SATO.

2. DECISIONS. In EDA formulae conflicts often express a correlation between variables that are close in terms of circuit connectivity. This suggests that the decision process should be biased towards unassigned variables that share clauses with already assigned variables. Our decision strategy has two components: *Local:* select the “best” variable that shares a clause with already assigned variables; and *Global:* select globally “best” variable. The mechanism of ranking variables in both groups is similar, though not identical, to that used in Chaff.

3. RESULTS. Table 1 summarizes results for the DIMACs benchmark suite. Each row presents benchmark name, and run time (in seconds) for GRASP, SATO, Chaff, and the new solver. A dash indicates that at least one benchmark test was timed-out. Table 2 summarizes results for the CMU Benchmark Suite. Note that all experiments reported in Tables 1 and 2 were run on the same 4CPU/400MHz/4GB Ultra-SPARC-4 machine. We used original unmodified GRASP, SATO, and zChaff packages with the following settings: GRASP(+T100 +B10000000 +C10000000 +S10000 +g20 +rt4 +dDLIS), SATO(default), and zChaff(default).

Table 1: DDIMACs benchmark results (in seconds).

name	GRASP	SATO	zChaff	new
ii16	-	2.18	67.71	**1.15**
ii32	4.47	2.47	2.75	**1.12**
ii8	2.03	0.48	0.53	**0.06**
aim200	5.81	0.77	1.11	**0.35**
aim100	0.70	0.20	0.20	**0.02**
ssa	3.30	3.78	0.28	**0.19**
jnh	5.69	1.04	0.72	**0.34**
dubois	0.35	0.17	0.14	**0.03**
hole	-	142.73	**42.76**	44.19
par16	-	-	**31.77**	39.26
hanoi 5	-	-	76705	**1186**

Table 2: CMU benchmark results (in seconds).

benchmark	zChaff	new	speed up
sss 1.0	99.14	**33.46**	2.96x
sss 1.0a	36.72	**19.79**	1.86x
sss-sat 1.0	671.75	**358.26**	1.88x
fvp-unsat-1.0	1480.24	**839.31**	1.76x
fvp-unsat-2.0	218268	**55081**	3.96x
vliw-sat 1.0	13626	**9236**	1.48x

References

[1] Joao P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Comput*, 48(5):506–520, May 1999.

[2] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proc.Design Automation Conference*, pages 530–535, June 2001.

[3] H. Zhang. SATO: An efficient propositional prover. In *Proc. of the International Conference on Automated Deduction*, pages 272–275, July 1997.