Sat-Solving Based on Boundary Point Elimination

Eugene Goldberg, Pete Manolios
Northeastern University, USA

HVC-2010, October 5-7, Haifa, Israel

Outline

- Introduction
- BPE-SAT
- Experimental results and conclusions

Motivation

Conflict Driven SAT-solvers with Clause Learning (CDCL-solvers) dominate the world of SAT-algorithms

However, they are still unrobust and unscalable (even for formulas with small proofs)

One of the main reasons: CDCL SAT-solvers do not take into account the formula structure

(some success with using formula symmetry).

Resolution as Boundary Point (BP) Elimination

BP ***p*** of F is a special kind of a complete assignment

F has no BPs $\Rightarrow F \equiv 0$

A BP ***p*** of F mandates a resolution (with resolvent C) such that : ***p*** is not a BP of $F \wedge C$

A resolution proof of $F \equiv 0$ can be viewed as **a process of BP elimination**

Completeness of Resolution Based on BP Elimination

Proof system based on elimination of BPs of the **original formula** F is, in general, **incomplete:**

One cannot always find a proof where every resolvent R_{k+1} eliminates a BP of $F \wedge R_1 \wedge .. \wedge R_k$

A complete proof system: every resolvent R_{k+1} has to eliminate **only** a BP of an **unsatisfiable subformula** of $F \wedge R_1 \wedge .. \wedge R_k$ (to be published)

Here, we consider the first case. Our approach can be extended to the second case too.

BP Elimination and Formula Structure

To find a **short** resolution proof a SAT-solver needs to find mandatory resolutions as soon as possible.

Finding a BP of the entire formula F is computationally hard.

An idea: find subformulas F' such that BPs of F' are also BPs of F.

Formula structure $\Leftrightarrow$ Knowledge of subformulas F' sharing BPs with F

BPE-SAT

- BPE-SAT is a template of resolution SAT-solvers.
- The main idea is to derive an empty clause from F by resolutions eliminating BPs of F
- BPs of F are eliminated by eliminating BPs of subformulas F'.

We conjecture that an efficient resolution SAT-solver has to be an implementation of BPE-SAT.

We show that a CDCL SAT-solver is a particular implementation BPE-SAT (and it is not always the best)

Outline

- Introduction
- BPE-SAT
- Experimental results and conclusions

Boundary Point (definition)

F – CNF formula, $\boldsymbol{p}$ – complete assignment, l is x or $\sim x$

$\boldsymbol{p}$ is l-BP of F if

$F(\boldsymbol{p})=0$ and

For all $C \in F$, $C(\boldsymbol{p}) = 0 \Rightarrow l \in C$

$\boldsymbol{p}'' = flip(\boldsymbol{p}', x)$

$\boldsymbol{p}'$ is l-BP of $F \Rightarrow$

$\boldsymbol{p}''$ is $\sim l$-BP of F or $F(\boldsymbol{p}'') = 1$

Elimination of $\boldsymbol{p}', \boldsymbol{p}'' = flip(\boldsymbol{p}', x)$:

Find $C'(\boldsymbol{p}')=0$ and $C''(\boldsymbol{p}'')=0$

Add resolvent C of C', C'' on x to F

Mandatory Resolutions

Let *F* be unsatisfiable and *l* be a literal of *x*.

- All BPs of *F* must be eliminated in a resolution proof
- An *l*-BP ***p*** of *F* is eliminated only by a resolution on variable *x* such that *C*(***p***)=0 where *C* is the resolvent

An *l*-BP **mandates** a particular resolution on *x*

Idea: use BPs to drive resolution proofs

A Key Point of BPE-SAT

Finding *l*-BP of *F* $\Rightarrow$ SAT-check for *F* \ {Clauses with *l*}

Generally, this is expensive

Idea: Eliminate *l*-BPs of ***subformulas*** of *F*

p is *l*-BP of *F* $\Rightarrow$

p is *l*-BP of any subformula *F′* of *F*, s.t. *F′* (***p***)=0

It is a conservative approach:

BPs eliminated from *F′* are not necessarily BPs of *F*

High-Level View of BPE-SAT

```
BPE-SAT(F)
  {while (true)
     {F′(X,Y)= Pick_subformula(F);
      R(Y) = Elim_vars(F′(X,Y));
      if (empty_clause ∈ R) return(Unsat);
     F = F ∪ R; }}
```

- *Elim_vars* quantifies away variables of *X*
- **To existentially quantify away $x \in X$ from F':**

$F' \Rightarrow F' \setminus \{\text{clauses of with } x\} \cup S$,

S eliminates all removable *x*-BPs, ~*x*-BPs of F'

BPE-SAT and CDCL Sat-Solvers

A CDCL SAT-solver is a special case of BPE-SAT

- Clauses responsible for a conflict $\Rightarrow$ subformula F'
- Conflict clause generation $\Rightarrow$ elimination of BPs of F'
- Eliminated BPs are trivially obtained from the assignment leading to the conflict

BPE paradigm provides explanation of why SAT-solvers based on DPLL are dominant.

A Flaw of CDCL SAT-Solvers

- Conflicts can not be controlled
- So subformulas F' corresponding to conflicts are "accidental"

BPs of F' eliminated by resolutions used to generate a conflict clause may not eliminate any BPs of F

- Let ***p*** be a BP of F' eliminated by resolving clauses of F' on x
- ***p*** may be falsified by a clause of $F \setminus F'$ that does not have variable x

Outline

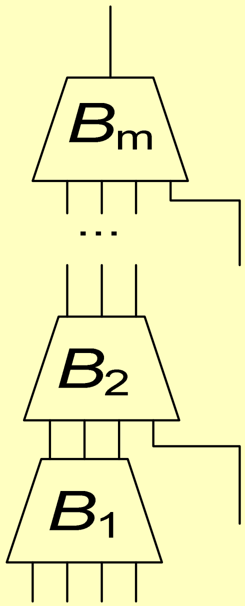
- Introduction
- BPE-SAT
- Experimental results and conclusions

Low Quality Proofs of a CDCL Sat-Solver

Solving formulas describing self-equivalence of *n*-bit multipliers

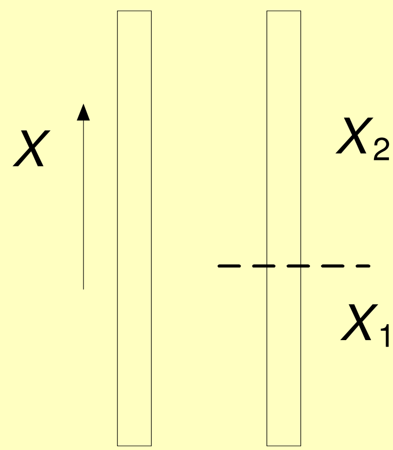
#bits	Picosat		Specialized proofs	
	#reso-lutions	mand. res. %	#reso-lutions	mand. res. %
2	215	77	77	100
3	2,958	66	409	100
4	32,117	38	957	100
5	231,270	24	1,697	100

Narrow Formulas *F*(*r*,*m*)



$N(r,m)$, $r = 3$

$F(r,m)$ describes equivalence checking of two implementations of $N(r,m)$.



F' consists of the clauses of F with at least one variable from X_1

elimination of BPs of F' $\Rightarrow$ elimination of BPs of F

Comparison of BPE-SAT and CDCL SAT-Solvers

Solving formulas $F(r,m)$. Time limit is 300,000 seconds.

(*r*,*m*)	**(*vars*, *cls*)** $\times 10^3$	***picosat* (s.)**	***bpe-sat* (s.)**	***ratio pico / bpe***
(3, 300)	(11, 30)	2.3	6.4	0.4
(3, 1000)	(35, 100)	69.6	30.8	2.2
(3, 2000)	(70, 200)	162	88.0	1.8
(4, 200)	(29, 86)	548	58.7	9.3
(4, 600)	(88, 258)	99,820	235	425
(4, 1000)	(147, 430)	**timeout**	955	> 314

Conclusions

- We formulate BPE-SAT, a template of resolution based SAT-solvers
- It is based on very general implications of the BP elimination concept
- CDCL SAT-solvers are a particular implementation of BPE-SAT
- We show the importance of knowing the formula structure (defined as subformulas sharing BPs with the formula)