Using Interval Constraint Propagation for Pseudo-Boolean Constraint Solving

Albert-Ludwigs-Universität Freiburg

Karsten Scheibler, Bernd Becker

Chair of Computer Architecture

FMCAD 2014

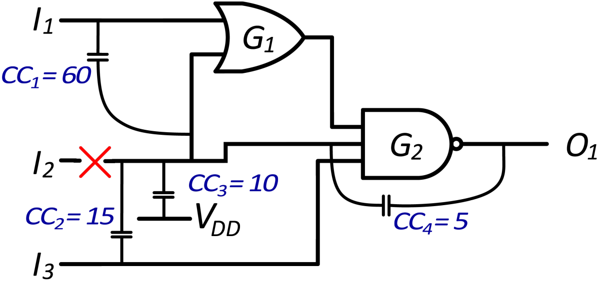
UNI FREIBURG

PB Constraints

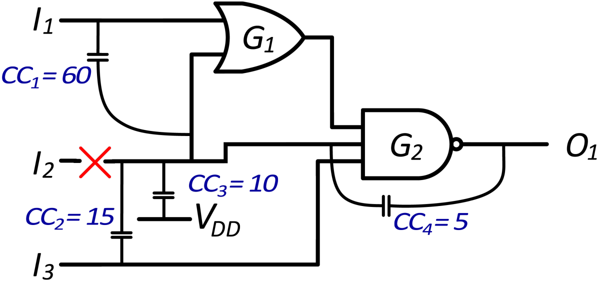
PB Constraints:

- $2x_1 + 4x_2 + x_3 < 5$
- $\sum_i c_i x_i \sim C$
- c_i and C integer coefficients, x_i boolean variables
- $\sim \in \{<, \leq, \geq, >\}$

ATPG for Open-Faults

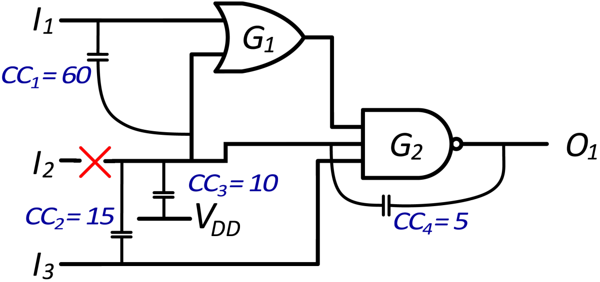


ATPG for Open-Faults



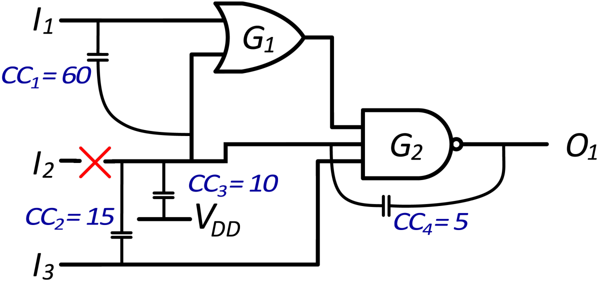
$$(O_1^{good} \Leftrightarrow \neg((I_1 \vee I_2^{good}) \oplus I_2^{good} \oplus I_3))\wedge$$

ATPG for Open-Faults



$$(O_1^{good} \Leftrightarrow \neg((I_1 \vee I_2^{good}) \oplus I_2^{good} \oplus I_3))\wedge$$
$$(I_2^{bad} \Leftrightarrow (60 \cdot I_1 + 15 \cdot I_3 + 5 \cdot O_1^{good} \geq 45))\wedge$$

ATPG for Open-Faults



$$(O_1^{good} \Leftrightarrow \neg((I_1 \vee I_2^{good}) \oplus I_2^{good} \oplus I_3)) \wedge$$
$$(I_2^{bad} \Leftrightarrow (60 \cdot I_1 + 15 \cdot I_3 + 5 \cdot O_1^{good} \geq 45)) \wedge$$
$$(O_1^{bad} \Leftrightarrow \neg((I_1 \vee I_2^{bad}) \oplus I_2^{bad} \oplus I_3)) \wedge (O_1^{good} \oplus O_1^{bad})$$

SAT Solvers, SMT Solvers

SAT solvers:

- e.g. $((a \oplus b) \vee (a \wedge \neg c))$

SMT solvers:

- SMT = SAT Modulo Theories
- e.g. $(\neg((x + y < 7) \vee (z^4 < 1)) \oplus (sin(x) \cdot z^2 = 3))$

$\rightsquigarrow$ SMT solver iSAT3 worked very well

SMT Solver iSAT3

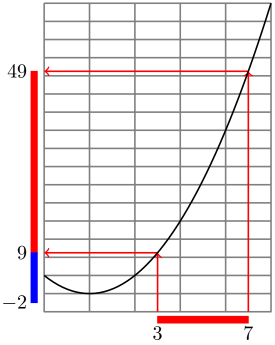
decomposing input formula into:

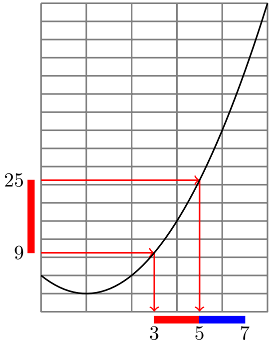
- simple bounds, e.g. $(h_1 < 7)$
- primitive constraints, e.g. $(h_5 = h_3 \cdot h_4)$

$$(\neg((x+y<7) \vee (z^4<1)) \oplus (sin(x) \cdot z^2 = 3))$$

$$(\neg((h_1<7) \quad \vee (h_2<1)) \oplus ((h_5 \geq 3) \wedge (h_5 \leq 3))) \wedge$$
$$(h_1 = x+y) \wedge (h_2 = z^4) \wedge (h_3 = sin(x)) \wedge (h_4 = z^2) \wedge (h_5 = h_3 \cdot h_4)$$

$h_4 = z^2,\ z \in [3,7],\ h_4 \in [-2,25]$





$z \in [3,7] \rightsquigarrow h_4 \geq 9$

$\rightsquigarrow \quad h_4 \in [9,25]$

$h_4 \in [9,25] \rightsquigarrow z \leq 5$

$\rightsquigarrow \quad z \in [3,5]$

Two Questions

1. Performance of other PB Solvers on our problem class?
   - Minisatp
   - SAT4JPB
   - Clasp

2. How well does iSAT3 perform on other PB benchmarks?
   - Benchmarks from PB competition 2012

The Answer

- PB competition 2012:
  - DEC-BIGINT-LIN = DBL, 14 benchmarks
  - DEC-SMALLINT-LIN = DSL, 355 benchmarks
  - DEC-SMALLINT-NLC = DSN, 30 benchmarks
- ATPG for Open-Faults = OF10, 321 benchmarks

Solver	DBL (14)	DSL (355)	DSN (30)	OF10 (321)	$\sum$
Minisatp	9	229	-	8	243
SAT4JPB	9	219	[10]	233	461 [471]
Clasp	5	234	[13]	287	526 [539]
iSAT3	2	155	[20]	313	470 [490]

Translation to SAT

A good translation

1. has small size
2. maintains GAC $(3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4 \quad x_3 \Rightarrow \neg x_1)$

	GAC	size
BDDs	++	-
Sorting networks	+	+
Adder cicuits	–	++

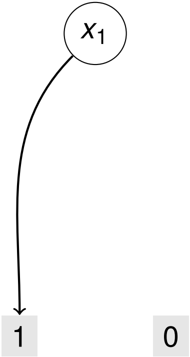
Translation to SAT with BDDs

$$3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4$$

1 0

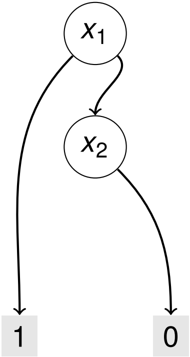
Translation to SAT with BDDs

$$3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4$$



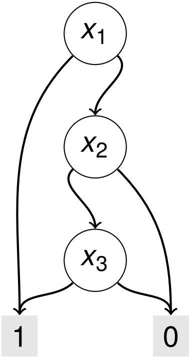
Translation to SAT with BDDs

$$3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4$$



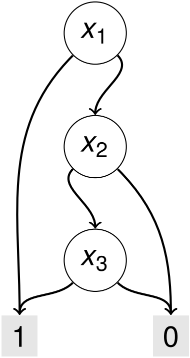
Translation to SAT with BDDs

$$3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4$$



Translation to SAT with BDDs

$$3 \cdot x_1 + 2 \cdot x_2 + x_3 < 4 \quad \rightsquigarrow \quad \neg x_1 \vee (x1 \wedge \neg x_2 \wedge \neg x_3)$$



Symbolic Gaussian Elimination

Basic idea: generate new lemmas from existing tautologies

Symbolic Gaussian Elimination

Basic idea: generate new lemmas from existing tautologies

$(y \geq 2.00001 \cdot x + 0.25) \wedge$ $\quad x, y \in [0, 1000000]$
$(y \leq 2 \cdot x)$

Symbolic Gaussian Elimination

Basic idea: generate new lemmas from existing tautologies

$(y \geq 2.00001 \cdot x + 0.25) \wedge$ $\qquad x, y \in [0, 1000000]$
$(y \leq 2 \cdot x)$

$(h_1 = y - 2.00001 \cdot x) \wedge (h_1 \geq 0.25) \wedge$ $\qquad x, y \in [0, 1000000]$
$(h_2 = y - 2 \cdot x) \wedge (h_2 \leq 0)$

Symbolic Gaussian Elimination

Basic idea: generate new lemmas from existing tautologies

$$(y \geq 2.00001 \cdot x + 0.25) \wedge \qquad x, y \in [0, 1000000]$$
$$(y \leq 2 \cdot x)$$

$$(h_1 = y - 2.00001 \cdot x) \wedge (h_1 \geq 0.25) \wedge \qquad x, y \in [0, 1000000]$$
$$(h_2 = y - 2 \cdot x) \wedge (h_2 \leq 0)$$

$$y - 2.00001 \cdot x - h_1 = 0$$
$$y - 2 \cdot x - h_2 = 0$$

replace y $\rightsquigarrow$ $0.00001 \cdot x + h_1 - h_2 = 0$ (add to formula)

Results

- DEC-BIGINT-LIN = DBL, 14 benchmarks
- DEC-SMALLINT-LIN = DSL, 355 benchmarks
- DEC-SMALLINT-NLC = DSN, 30 benchmarks
- OPENFAULTS-DIV10 = OF10, 321 benchmarks

Solver	DBL (14)	DSL (355)	DSN (30)	OF10 (321)	$\sum$
Minisatp	9	229	-	8	243
SAT4JPB	9	219	[10]	233	461 [471]
Clasp	5	234	[13]	287	526 [539]
iSAT3	2	155	**[20]**	313	470 [490]
iSAT3 + BDD	**14**	208	**[20]**	**315**	537 [557]
iSAT3 + BDD + SGE	**14**	**238**	**[20]**	**315**	**567 [587]**

Conclusion & Outlook

Conclusion:

- SAT-Translations of large PB Constraints may perform poor
- Advantageous to have an arithmetic reasoning mechanism as fall-back

Outlook:

- Generating lemmas during solving